whose orbit approaches L as $k \to \infty$. The stable manifold is invariant under P.

- *Unstable manifold (discrete-time)*: The unstable manifold $W^u(L)$ of a limit set L of a diffeomorphism P is the set of all points x whose orbit approaches L as $k \to -\infty$. The unstable manifold is invariant under P.

- *Homoclinic point*: Let x^* be a fixed point of a diffeomorphism P. A homoclinic point is a point $x \neq x^*$ that lies in $W^s(x^*) \bigcap W^u(x^*)$. The existence of one homoclinic point implies the existence of an infinity of homoclinic points.

- *Homoclinic orbit*: An orbit composed of homoclinic points.

- *Transversal homoclinic point*: A homoclinic point at which the manifolds intersect transversally. A map that possesses a transversal homoclinic point has horseshoe-like maps embedded in it and, therefore, exhibits sensitive dependence on initial conditions.

Chapter 7

Dimension

This chapter addresses the question of the dimension of a limit set, in particular, the dimension of a strange attractor. We will see that a strange attractor possesses non-integer dimension while the dimension of a non-chaotic attractor is always an integer.

After we discuss dimension, we present a remarkable result that permits an attractor to be reconstructed from a sampled time waveform of just one component of the state.

7.1 Dimension

There are several different types of dimension. The dimension of Euclidean space is familiar to everyone—it is the minimum number of coordinates needed to specify a point uniquely. The dimension of a dynamical system is the number of state variables that are used to describe the dynamics of the system. In differential topology, the dimension of a manifold is the dimension of the Euclidean space that the manifold resembles locally. None of these dimensions allows non-integer values and none of them can be used to describe strange attractors. The generic term for a dimension that allows non-integer values is *fractal dimension.* A set that has non-integer dimension is called a *fractal.* Almost all strange attractors are fractals.

We present five different types of fractal dimension. The most well-known is capacity. The four others are information dimension, correlation dimension, kth nearest-neighbor dimension, and Lyapunov dimension.

Though these five are the most commonly used dimensions, there are several other definitions. Some enlightening discussions on these

other dimensions and their relationship to the ones presented in this chapter can be found in Young [1983], Farmer *et al.* [1983], Badii and Politi [1985], and Mayer-Kress [1986].

7.1.1 Definitions

Capacity

The simplest type of dimension is *capacity.* Cover an attractor A with volume elements (spheres, cubes, etc.) each with diameter ϵ. Let $N(\epsilon)$ be the minimum number of volume elements needed to cover A. If A is a D-dimensional manifold—D is necessarily an integer—then the number of volume elements needed to cover A is inversely proportional to ϵ^D, that is,

$$N(\epsilon) = k\,\epsilon^{-D} \tag{7.1}$$

for some constant k.[1] The capacity, denoted by D_{cap}, is obtained by solving (7.1) for D and taking the limit as ϵ approaches zero,

$$D_{cap} := \lim_{\epsilon \to 0} \frac{\ln N(\epsilon)}{\ln(1/\epsilon)}. \tag{7.2}$$

If the limit does not exist, then D_{cap} is undefined. Since a d-dimensional manifold locally resembles $\mathbb{R}^d$, D_{cap} of a manifold equals the topological dimension, which is an integer. For objects that are not manifolds, D_{cap} can take on non-integer values.

Example 7.1 The unit interval:
As volume elements, choose intervals of length $\epsilon = 1/3^k$. It takes $N(\epsilon) = 3^k$ of these volume elements to cover the unit interval $[0, 1]$ (see Fig. 7.1(a)). To refine the covering, let $k \to \infty$ to obtain

$$D_{cap} = \lim_{k \to \infty} \frac{\ln 3^k}{\ln 3^k} = 1. \tag{7.3}$$

As expected, the unit interval has dimension 1.

Example 7.2 The middle-third Cantor set:
This example will show that the middle-third Cantor set has non-integer dimension. Since strange attractors have a Cantor-set-like

[1] k depends on the geometry of the attractor and on the type of volume element used.

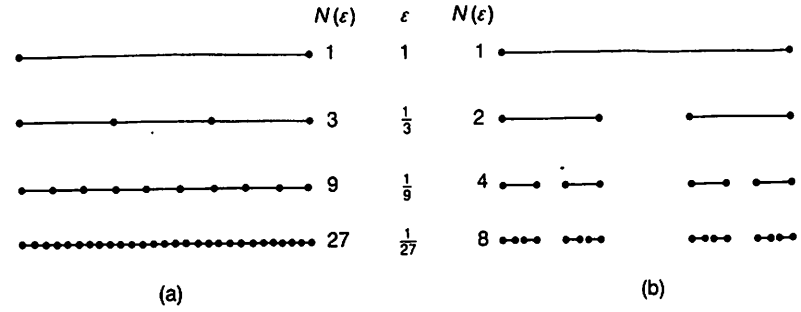


Figure 7.1: Two simple examples of capacity. (a) The unit interval; (b) the middle-third Cantor set.

structure, the example also explains why a strange attractor has a non-integer dimension.

The middle-third Cantor set is constructed iteratively by removing the middle third of the unit interval and then the middle third of the remaining two intervals, etc. (see Appendix F). To calculate the capacity of the middle-third Cantor set, cover it, at the kth step in the construction, with intervals of length $\epsilon = 1/3^k$ as shown in Fig. 7.1(b). At the kth step, the number of intervals required to cover the set is $N(\epsilon) = 2^k$, and

$$D_{cap} = \lim_{k \to \infty} \frac{\ln 2^k}{\ln 3^k} = \frac{\ln 2}{\ln 3} = 0.6309\ldots. \tag{7.4}$$

Hence, the Cantor set is something more than a point (dimension 0) but something less than an interval (dimension 1).

Remark: When the limit in (7.2) exists, the question arises whether another covering (e.g., spheres instead of cubes or, perhaps, a mixture of spheres of different sizes) can result in a different value of D_{cap}. The answer, unfortunately, is yes. To resolve this dilemma, we simply comment that capacity is closely related to Hausdorff dimension and that the definition of Hausdorff dimension implies that if different coverings result in different values of D_{cap}, then the minimum value over all coverings should be used (see Young [1983]).

Information dimension

Capacity is a purely metric concept. It utilizes no information about the time behavior of the dynamical system. Information dimension,

on the other hand, is defined in terms of the relative frequency of visitation of a typical trajectory. The setting is the same as for capacity—a covering of $N(\epsilon)$ volume elements each with diameter ϵ. The *information dimension* D_I is defined by

$$D_I := \lim_{\epsilon \to 0} \frac{H(\epsilon)}{\ln(1/\epsilon)} \tag{7.5}$$

where

$$H(\epsilon) := -\sum_{i=1}^{N(\epsilon)} P_i \ln P_i. \tag{7.6}$$

P_i is the relative frequency with which a typical trajectory enters the ith volume element of the covering.

Readers familiar with information theory will recognize $H(\epsilon)$ as entropy—the amount of information needed to specify the state of the system to an accuracy of ϵ if the state is known to be on the attractor.

For sufficiently small ϵ, (7.5) can be rewritten as

$$H(\epsilon) = k\epsilon^{-D_I} \tag{7.7}$$

for some constant of proportionality k. In words, the amount of information needed to specify the state increases inversely with the D_Ith power of ϵ. Compare this equation with (7.1).

Example 7.3 Unit interval:
Assume the attractor is the unit interval and that the probability density is uniform. As before, choose intervals of length $\epsilon = 1/3^k$ as volume elements. It follows that $N(\epsilon) = 3^k$ and that $P_i = 1/3^k$. With these values, the entropy is

$$\begin{aligned} H(\epsilon) &= -\sum_{i=1}^{3^k} \frac{1}{3^k} \ln \frac{1}{3^k} \\ &= \ln 3^k \end{aligned} \tag{7.8}$$

and the information dimension is

$$D_I = \lim_{k \to \infty} \frac{\ln 3^k}{\ln 3^k} = 1. \tag{7.9}$$

which agrees with D_{cap}. In fact, it is easy to show that for uniform densities, D_{cap} and D_I always agree.

Example 7.4 Interval and point:
Let the set under study be the subset of $\mathbb{R}$ that consists of the unit interval $[0, 1]$ and the isolated point 2. Assume that the probability of finding a point at 2 is $1/2$, and of finding a point on the interval is also $1/2$. Further assume that the density is uniform over the interval.

It is easy to show that the capacity of this set is 1. This is the same result as for the unit interval itself and, therefore, capacity ignores the isolated point even though the probability of being at the point is the same as being on the interval.[2]

To find the information dimension of this set, choose intervals of length $\epsilon = 1/3^k$ as volume elements. It takes 3^k such elements to cover the unit interval but only one to cover the isolated point, so $N(\epsilon) = 3^k + 1$. $P_i = 1/2$ for the volume element covering the isolated point and $P_i = 1/(2 \cdot 3^k)$ for the remaining volume elements. Thus,

$$\begin{aligned} H(\epsilon) &= -\frac{1}{2}\ln\left(\frac{1}{2}\right) - \frac{1}{2}\sum_{i=1}^{3^k} \frac{1}{3^k} \ln\left(\frac{1}{2 \cdot 3^k}\right) \\ &= \frac{1}{2}\ln 2 + \frac{1}{2}\ln(2 \cdot 3^k) \\ &= \ln 2 + \frac{1}{2}\ln 3^k \end{aligned} \tag{7.10}$$

and

$$\begin{aligned} D_I &= \lim_{k \to \infty} \left(\frac{\ln 2}{\ln 3^k} + \frac{\ln 3^k}{2 \ln 3^k} \right) \\ &= \frac{1}{2}. \end{aligned} \tag{7.11}$$

Thus, for this example, D_I is the average of the dimensions of the point and of the interval.

Though the set in the last example is not the limit set of a dynamical system, it does show how information dimension differs from capacity. Capacity tends to ignore lower dimensional subsets of the attractor. Information dimension, on the other hand, weights the lower-dimensional subsets according to the frequency of visitation of a typical trajectory.

Example 7.5 The middle-third Cantor set:
To find the information dimension of the middle-third Cantor set,

[2] It can be shown using (7.1) and (7.2) that $D_{cap}(S) = \max_i D_{cap}(S_i)$ for $S = S_1 \bigcup \cdots \bigcup S_k$ for some finite k.

it is necessary to define a probability density on the set. In each step of the construction of the Cantor set, the middle of an interval is removed leaving two smaller sub-intervals, one on the left and one on the right. Any point in the Cantor set is, therefore, identified uniquely by its left-right history, that is, whether at the kth step of the construction it was in the right or left sub-interval. For example, $\ell\ell\ell\ell\ldots$ is the left-most point of the Cantor set (i.e., 0), and $\ell rr\ldots$ is the right-most point of the first left sub-interval (i.e., 1/3).

The left-right history can be used to define a self-similar probability density. Let $0 \le p_\ell \le 1$ be the probability of being in the left sub-interval and $p_r = 1 - p_\ell$ be the probability of being in the right sub-interval. For example, the probability that a point lies in the segment [2/27, 3/27] corresponding to $\ell\ell r$ is $p_\ell p_\ell p_r$. It is shown in Appendix F that the information dimension of the middle-third Cantor set with this probability density is

$$D_I = -\frac{p_\ell \ln p_\ell + p_r \ln p_r}{\ln 3}. \tag{7.12}$$

Observe that when $p_\ell = p_r = 1/2$, the information dimension agrees with the capacity. When $p_\ell \neq p_r$, however, the information capacity is always less than the capacity.

Remark: It can be shown that for any attractor, $D_I \le D_{cap}$.

Correlation dimension

Another probabilistic type of dimension is the *correlation dimension* D_C. It, too, depends upon refining a covering of $N(\epsilon)$ volume elements of diameter ϵ, and is defined by

$$D_C := \lim_{\epsilon\to 0} \frac{\ln \sum_{i=1}^{N(\epsilon)} P_i^2}{\ln \epsilon} \tag{7.13}$$

where, as before, P_i is the relative frequency with which a typical trajectory enters the ith volume element.

To help interpret the numerator of (7.13), suppose N points of a trajectory have been collected, either through simulation or from measurements. Define the correlation as

$$C(\epsilon) := \lim_{N\to\infty} \frac{1}{N^2}\{\textit{the number of pairs of points } (x_i, x_j) \textit{ such that } \|x_i - x_j\| < \epsilon\}. \tag{7.14}$$

Then

$$D_C = \lim_{\epsilon\to 0} \frac{\ln C(\epsilon)}{\ln \epsilon}. \tag{7.15}$$

To show the plausibility of (7.15), let n_i be the number of points lying in the ith volume element. Then

$$P_i = \lim_{N\to\infty} \frac{n_i}{N}. \tag{7.16}$$

Since the volume element has diameter ϵ, all the n_i points lie within ϵ of each other and form $n_i^2 - n_i$ pairs of points.[3] It follows that

$$\begin{aligned} C(\epsilon) &= \lim_{N\to\infty} \frac{1}{N^2} \sum_{i=1}^{N(\epsilon)} (n_i^2 - n_i) \\ &= \sum_{i=1}^{N(\epsilon)} \lim_{N\to\infty} \frac{n_i^2}{N^2} - \lim_{N\to\infty} \frac{n_i}{N^2} \\ &= \sum_{i=1}^{N(\epsilon)} P_i^2 - \lim_{N\to\infty} \frac{P_i}{N} \\ &= \sum_{i=1}^{N(\epsilon)} P_i^2 \end{aligned} \tag{7.17}$$

from which (7.15) follows.

An objection to the preceding derivation can be made because $n_i^2 - n_i$ is the number of points in a *single* volume element. It does not include pairs of points that are within ϵ of each other but that lie in different volume elements. In response to this objection, assume that there are actually μn_i^2 pairs of points within ϵ of one another where $\mu > 1$ is a correction factor. With this correction term, (7.15) becomes

$$\begin{aligned} D_C &= \lim_{\epsilon\to 0} \frac{\ln(\mu C(\epsilon))}{\ln \epsilon} \\ &= \lim_{\epsilon\to 0} \frac{\ln \mu}{\ln \epsilon} + \lim_{\epsilon\to 0} \frac{\ln C(\epsilon)}{\ln \epsilon} \\ &= \lim_{\epsilon\to 0} \frac{\ln C(\epsilon)}{\ln \epsilon} \end{aligned} \tag{7.18}$$

which agrees with (7.15).

[3] (x_i, x_i) is not counted as a pair and for $i \neq j$, (x_i, x_j) is treated as a pair different from (x_j, x_i).

Example 7.6 Interval and point:
Consider the set from Example 7.4. Use intervals of length $\epsilon = 1/3^k$ as volume elements. Then $N(\epsilon) = 3^k + 1$, $P_i = 1/(2 \cdot 3^k)$ for the volume elements covering the interval, and $P_i = 1/2$ for the single volume element covering the isolated point. Substitute these values into the numerator of (7.13) to obtain

$$\begin{aligned} \ln \sum_{i=1}^{N(\epsilon)} P_i^2 &= \ln \left[\frac{1}{4} + \sum_{i=1}^{3^k} \frac{1}{4 \cdot 3^{2k}} \right] \\ &= \ln \left[\frac{1}{4} \left(1 + \frac{1}{3^k} \right) \right]. \end{aligned} \tag{7.19}$$

which yields

$$\begin{aligned} D_C &= \lim_{k \to \infty} \frac{\ln \left[\frac{1}{4} \left(1 + \frac{1}{3^k} \right) \right]}{\ln(1/3^k)} \\ &= \lim_{k \to \infty} \frac{\ln(1/4)}{\ln(1/3^k)} \\ &= 0. \end{aligned} \tag{7.20}$$

Thus, unlike capacity and information dimension, the correlation dimension ignores the unit interval entirely. The reader may find it interesting to calculate D_C for this example using (7.15).

Example 7.7 The middle-third Cantor set:
Consider the middle-third Cantor set with probabilities as in Example 7.5. It is shown in Appendix F that the correlation dimension of the middle-third Cantor set is

$$D_C = -\frac{\ln(p_l^2 + p_r^2)}{\ln 3}. \tag{7.21}$$

Observe that when $p_l = p_r = 1/2$, the correlation dimension agrees with the information dimension and with the capacity. When, however, $p_l \neq p_r$, the correlation dimension is always less than the information dimension.

Remark: It can be shown that for any attractor, $D_C \leq D_I \leq D_{cap}$.

kth nearest-neighbor dimension

The kth nearest-neighbor dimension was formulated by Pettis *et al.* [1979]. The appeal of this dimension is that its definition is based firmly on probabilistic concepts.

Consider an attractor A embedded in $\mathbb{R}^n$. Let $x_1, \ldots, x_N$ be N randomly chosen data points lying on A. Let $r(k,x)$ be the distance between x and its kth nearest neighbor in $\{x_i\}$. Define $\bar{r}(k)$ as the mean of $r(k,x)$ taken over $\{x_i\}$, that is,

$$\bar{r}(k) := \frac{1}{N} \sum_{i=1}^{N} r(k, x_i). \tag{7.22}$$

Pettis *et al.* show that under reasonable assumptions and for large N, there exist functions g and c such that the *kth nearest-neighbor dimension* D_{nn} is well-defined by the equation

$$D_{nn} := \frac{\ln k + c(x_1, \ldots, x_N)}{g(k, D_{nn}) + \ln \bar{r}(k)}. \tag{7.23}$$

Pettis *et al.* show that $g(k, D_{nn})$ is small for all k and D_{nn}, so ignoring g, (7.23) can be rewritten as

$$\bar{r}(k) \approx e^{c(x_1, \ldots, x_N)} k^{1/D_{nn}} \tag{7.24}$$

which, given $\{x_1, \ldots, x_N\}$, implies that the average distance to the kth nearest neighbor is proportional to $k^{1/D_{nn}}$. This proportionality is intuitively satisfying, at least for manifolds. For example, let $\{x_1, \ldots, x_N\}$, be chosen from the interior of the unit circle using a uniform probability density. Let S be a smaller circle with radius r that is randomly positioned in the interior of the unit circle. The number $n(r)$ of points in the intersection of S and $\{x_i\}$ is, on the average, proportional to the area of S, that is,

$$n(r) \propto r^2. \tag{7.25}$$

Since a circle of radius $\bar{r}(k)$ contains, on the average, k points, it follows from (7.25) that $k \propto \bar{n}(r)^2$ which agrees with (7.24).

Owing to the complexity of calculating $\bar{r}(k)$, simple yet meaningful analytical examples are difficult to find. Pettis *et al.* give an example using a uniform density over the unit interval with $N = 3$.

Lyapunov dimension

Let $\lambda_1 \geq \cdots \geq \lambda_n$ be the Lyapunov exponents of an attractor of a continuous-time dynamical system. Let j be the largest integer such that $\lambda_1 + \cdots + \lambda_j \geq 0$. The *Lyapunov dimension* as defined by Kaplan and Yorke [1979] is

$$D_L := j + \frac{\lambda_1 + \cdots + \lambda_j}{|\lambda_{j+1}|}. \tag{7.26}$$

If no such j exists, as is the case for a stable hyperbolic equilibrium point, D_L is defined to be 0.

For an attractor, $\sum_{i=1}^{n} \lambda_i < 0$, so j is guaranteed to be less than n. For an attracting limit cycle, the generic situation is $\lambda_1 = 0 > \lambda_2 > \cdots > \lambda_n$. Thus, the Lyapunov dimension of a generic attracting limit cycle is 1. Similarly, the Lyapunov dimension of generic attracting K-periodic behavior is K.

If the attractor is chaotic, D_L is almost always a non-integer.[4] In a three-dimensional chaotic system with Lyapunov exponents $\lambda_+ > 0 > \lambda_-$,

$$D_L = 2 + \frac{\lambda_+}{|\lambda_-|}. \tag{7.27}$$

For an attractor, $\lambda_+ + \lambda_- < 0$ from which it follows that $2 < D_L < 3$.

Plausibility argument: The derivation of (7.26) by Kaplan and Yorke is not rigorous, and we present a plausibility argument. Consider an n-dimensional hyper-cube C evolving in a flow ϕ_t. Let the length of the sides of the hyper-cube be ϵ. With the proper change of coordinates and for ϵ small, the ith side of C evolves under the flow, on the average, as $\epsilon e^{\lambda_i t}$. To find the capacity of C cover C at time 0 with hyper-cubes of side ϵ. To refine this covering, let each of the sides of each of the volume elements contract at the constant rate $e^{\lambda_{k+1} t}$ where k is chosen such that $\lambda_{k+1} < 0$ (the λ_i are in decreasing order as before). The number of these contracting cubes it takes to cover C at time t is

$$\begin{aligned} N(t) &= \frac{\epsilon e^{\lambda_1 t}}{\epsilon e^{\lambda_{k+1} t}} \cdots \frac{\epsilon e^{\lambda_k t}}{\epsilon e^{\lambda_{k+1} t}} \\ &= e^{(\lambda_1 + \cdots + \lambda_k - k\lambda_{k+1})t}. \end{aligned} \tag{7.28}$$

[4] In the non-generic case of $\lambda_1 > 0$, $\lambda_2 = 0$, $\lambda_3 = -\lambda_1$, and $\lambda_4 < \lambda_3$, the attractor has Lyapunov dimension 3.

The sides of C that grow with rates $\lambda_{k+2}, \ldots, \lambda_n$, do not influence $N(t)$ because they are shrinking with respect to $e^{\lambda_{k+1} t}$.

The capacity is approximated by

$$\begin{aligned} D_{cap}(k) &= -\lim_{t \to \infty} \frac{\ln N(t)}{\ln(\epsilon e^{\lambda_{k+1} t})} \\ &= k - \frac{\lambda_1 + \cdots + \lambda_k}{\lambda_{k+1}}. \end{aligned} \tag{7.29}$$

It is shown by Frederickson *et al.* [1983] that $k = j$ yields the lowest value[5] of $D_{cap}(k)$ so define

$$\begin{aligned} D_L := D_{cap}(j) &= j - \frac{\lambda_1 + \cdots + \lambda_j}{\lambda_{j+1}} \\ &= j + \frac{\lambda_1 + \cdots + \lambda_j}{|\lambda_{j+1}|} \end{aligned} \tag{7.30}$$

to complete the plausibility argument.

Discussion

Given the different definitions of dimension, it is natural to ask what relationship they bear to one another. Are they equivalent? Is one more useful than another?

First, we warn the reader that dimension is an active research area and the relationships and meanings of the different dimensions are unclear, especially in experimental settings or when applied to simulations. Part of the problem arises from not having an exact definition of a strange attractor. Other problems are due to the difficulty of analyzing the statistical properties that are required for D_I, D_C, D_{nn}, and D_L. Until these issues are settled, it is difficult to make any rigorous statements regarding the dimension of a strange attractor. Thus, this discussion is necessarily speculative in nature.

One task for which dimension appears to be impractical is to describe the geometric structure of an attractor. It seems ridiculous that a single number can fully describe the complex structure of a strange attractor. It is shown in Appendix F that D_{cap}, D_I, and D_C are three members of a countably infinite family of dimensions called the Renyi dimensions. Moreover, Badii and Politi [1985] introduced

[5] Recall that if two different coverings result in different capacities, the minimum value should be used.

an entire continuum of dimensions in which, at least for self-similar sets, the Renyi dimensions are embedded. A different continuum of dimensions, the *multifractal spectrum*, has been used to measure apparently universal properties of attractors at the onset of chaos through period-adding bifurcations (see Glazier and Libchaber [1988] for a review of this work). The current state of knowledge is unclear as to how much information these different values of dimension carry. For example, to distinguish between two attractors on the basis of dimension alone, is it sufficient to know the values of all these different dimensions, of just a few, or is additional information needed?

Dimension can be used to classify strange versus non-strange attractors. Non-strange attractors have integer dimension, and almost all chaotic attractors have non-integer dimension. This classification scheme, though nice in theory, is fairly useless in practice. First, the dimension algorithms have low precision, making it difficult if not impossible to judge whether the dimension is an integer. Second, there are better ways to judge whether an attractor is chaotic (e.g., looking at it, or one of the shooting methods).

The main use of dimension is to quantify the complexity of an attractor. The dimension of an attractor gives a lower bound on the number of state variables needed to describe the dynamics on the attractor. In words familiar from physics, the dimension is a lower bound on the number of degrees of freedom of the attractor. For example, motion on a limit cycle (dimension 1) can be described by a first-order differential equation where the variable is arc-length along the circle or perhaps the angle of rotation. In the chaotic case, motion on an attractor with dimension 2.4 cannot be described by two state variables, but can be modeled, at least theoretically, by a third-order system.

If the main reason for finding the dimension of an attractor is to estimate the minimum number of variables needed to describe the steady-state dynamics, there seems to be no theoretical reason for choosing one type of dimension over another—D_{cap}, D_I, D_C, and D_{nn} give values close to one another. The main reasons for choosing one type of dimension over another are the ease and accuracy of its computation. Since new algorithms may be found, there is no cut and dried answer as to which dimension-finding algorithm is best. There exist algorithms of comparable efficiency for finding each of D_{cap}, D_C, and D_{nn}. These algorithms are presented in the next section.

As is obvious from the definition, Lyapunov dimension is somewhat different from the other four. It was originally conjectured that $D_L = D_{cap}$ as the plausibility argument is designed to show. Since Lyapunov exponents are probabilistic in nature—they are defined in terms of an average over time—and D_{cap} is not, the conjecture was changed to $D_L = D_I$. Numerical simulations do not appear to contradict this conjecture, but analytical examples exist where $D_L \neq D_I$ (see Grassberger and Procaccia [1983]). These examples, however, are not structurally stable and if perturbed, the relationship $D_L = D_I$ does hold. The exact relationship between D_L and D_I (and the other dimensions as well) is an active research topic.

7.1.2 Algorithms

For each type of dimension, there are several different algorithms for estimating it. We cannot cover all the different techniques and are satisfied to present one technique for each of D_{cap}, D_C, and D_{nn}. This is an active field of research and the algorithms presented here should not be taken as the best possible algorithms—most likely they will be superseded by more reliable algorithms in the near future. The algorithms presented do demonstrate, however, the wide variety of approaches that are available to tackle the problem of estimating dimension and they highlight the creativity and energy researchers have devoted to this area.

The input to the following dimension-finding algorithms is a finite sequence $\{x_1, \ldots, x_N\}$ of points on an attractor. Typically, the points are evenly spaced time-samples of one or more trajectories that have achieved the steady state. It is important to keep in mind that the dimension-finding algorithms are not given full information about an attractor; they are given only partial information about a finite number of trajectories on the attractor.

Capacity

The first attempts to calculate capacity were based directly on the definition. From (7.1),

$$\ln N(\epsilon) = D_{cap} \ln \frac{1}{\epsilon} + \ln k. \tag{7.31}$$

Hence, D_{cap} is the slope of a log-log plot of $N(\epsilon)$ versus $1/\epsilon$.

To calculate $N(\epsilon)$ for several values of ϵ simultaneously, choose some minimum value ϵ_0 of ϵ. Cover the n-dimensional state space with a grid of boxes (hyper-cubes) of side ϵ_0. In the computer, maintain a boolean array—all entries initialized to FALSE—where each entry corresponds to one of the boxes. For each data point x_i, calculate the indices of the box in which x_i lies and set the corresponding entry of the array to TRUE. $N(\epsilon_0)$ is the number of array entries that are TRUE. For $m = 2, \ldots, M$, the value $N(2^m\epsilon_0)$ can be calculated from the entries of the boolean array by partitioning the ϵ_0 grid into larger cubes of side $\epsilon_0 2^m$. $N(\epsilon_0 2^m)$ is the number of these larger cubes that contain at least one ϵ_0 cube with a TRUE boolean array entry. Note that the choice of $\epsilon = \epsilon_0 2^m$ yields evenly spaced points on the x-axis of the log-log plot.

This box-counting technique works, but has two drawbacks.

1. For systems of dimension greater than three, the memory requirements are excessive. The total number of entries in the boolean array is proportional to $(1/\epsilon_0)^n$. For $\epsilon_0 = 0.01$ and $n = 4$, there are on the order of 10^8 entries in the array.

2. A large amount of data is required to ensure that nearly every ϵ_0 cube that contains a point on the attractor is visited by the trajectory under study; otherwise, the estimate of D_{cap} will be low. This point is directly related to the fact that capacity does not distinguish between those parts of the attractor that are visited frequently and those parts that are rarely touched by a typical trajectory.

A more efficient algorithm for estimating D_{cap} was presented by Hunt and Sullivan [1986]. Given an attractor A embedded in $\mathbb{R}^n$, define

$$\bar{A}(\epsilon) := \text{volume}\,\{y : \text{dist}\,(y, A) < \epsilon\} \tag{7.32}$$

where

$$\text{dist}\,(y, A) := \inf_{x \in A} \|x - y\| \tag{7.33}$$

is the distance from y to A.

Let D_{cap} be the capacity of A and define

$$D_A := \lim_{\epsilon \to 0} \frac{\ln(\bar{A}(\epsilon))}{\ln \epsilon}. \tag{7.34}$$

We now show that

$$D_{cap} = n - D_A \tag{7.35}$$

and, therefore; that D_A can be used to calculate D_{cap}.

Cover A with $N(\epsilon)$ hyper-cubes of side ϵ. Every point y that is within ϵ of A lies in one of these $N(\epsilon)$ cubes or in a cube that neighbors one of these cubes. Each cube has $3^n - 1$ neighbors so

$$\bar{A}(\epsilon) \leq 3^n \epsilon^n N(\epsilon). \tag{7.36}$$

Take the log of both sides and divide by $\ln \epsilon$ (which is negative for ϵ small) to obtain

$$\frac{\ln \bar{A}(\epsilon)}{\ln \epsilon} \geq \frac{n \ln 3}{\ln \epsilon} + \frac{n \ln \epsilon}{\ln \epsilon} + \frac{\ln N(\epsilon)}{\ln \epsilon}. \tag{7.37}$$

Take the limit as $\epsilon \to 0$ to obtain

$$D_A \geq n - D_{cap}. \tag{7.38}$$

Now cover A with $N(\epsilon/\sqrt{n})$ hyper-cubes of side $\epsilon/\sqrt{n}$. Any point in these cubes is within ϵ of A so

$$\bar{A}(\epsilon) \geq \left(\frac{\epsilon}{\sqrt{n}}\right)^n N(\epsilon/\sqrt{n}). \tag{7.39}$$

Take the log of both sides and divide by $\ln(\epsilon/\sqrt{n})$ to obtain

$$\frac{\ln \bar{A}(\epsilon)}{\ln \epsilon - \ln \sqrt{n}} \leq \frac{n \ln(\epsilon/\sqrt{n})}{\ln(\epsilon/\sqrt{n})} + \frac{\ln N(\epsilon/\sqrt{n})}{\ln(\epsilon/\sqrt{n})}. \tag{7.40}$$

Take the limit as $\epsilon \to 0$ to obtain

$$D_A \leq n - D_{cap}. \tag{7.41}$$

Equation (7.35) follows from (7.38) and (7.41).

The advantage of D_A over D_{cap} is that D_A can be calculated more efficiently than D_{cap}. D_A is calculated by finding the slope of a log-log plot of $\bar{A}(\epsilon)$ versus ϵ. $\bar{A}(\epsilon)$ can be calculated efficiently using Monte Carlo techniques. It is not possible to present a full explanation of Monte Carlo techniques here, so we present only a brief description. Normalize the coordinates such that A sits in an n-dimensional unit hyper-cube in $\mathbb{R}^n$. Generate K random points $x_1, \ldots, x_K$ in this hyper-cube. Count the number N_A of these points that lie within ϵ of A. Then $\bar{A}(\epsilon) \approx N_A/M$. The most costly step in this technique is calculating whether each x_i lies within ϵ of A. Hunt and Sullivan [1986] present a tree-like data structure that implements this Monte Carlo technique efficiently on vector floating-point processors.

Correlation dimension

The correlation dimension can be found directly from (7.13) using a box-counting scheme to estimate P_i, but as explained earlier, this approach is inefficient. A faster algorithm, due to Grassberger and Procaccia [1983], is obtained using the correlation $C(\epsilon)$ defined in (7.14). From (7.15), the correlation dimension is the slope of a log-log plot of $C(\epsilon)$ versus ϵ. For a given ϵ, estimating $C(\epsilon)$ entails computing all the inter-point distances, $r_{ij} := \|x_i - x_j\|$, and counting the number $N_r(\epsilon)$ of $r_{ij} < \epsilon$ for $i, j = 1, \ldots, N$. Then, $C(\epsilon) = N_r(\epsilon)/N^2$.

The x-axis of the log-log plot is $\ln \epsilon$. To obtain a set of points that are evenly spaced in the x direction, a simple binning algorithm is used. Calculate $N(\epsilon)$ for values of ϵ that are geometrically spaced, that is, for ϵ_0, ϵ_0^2, ..., ϵ_0^K, for some $\epsilon_0 > 0$ and some integer $K > 0$. This task is best accomplished by maintaining a K-dimensional array $N_\epsilon[\,]$ of integers. $N_\epsilon[k]$ is the count of the inter-point distances that satisfy $\epsilon_0^{k-1} < r_{ij} < \epsilon_0^k$. Then

$$N(\epsilon_0^k) = \sum_{i=1}^{k} N_\epsilon[i], \qquad k = 1, \ldots, K. \tag{7.42}$$

The $N_\epsilon[\,]$ array is initialized to all zeros. The most straightforward way to fill $N_\epsilon[\,]$ is to choose ϵ_0 equal to some small integer b. For each of the inter-point distances r_{ij}, set k to the integer part of $\log_b r_{ij}$ and then increment $N_\epsilon[k]$ by one.[6] The drawback of this approach is that log() is an expensive floating-point operation.

The $N_\epsilon[\,]$ array can be filled more efficiently by taking advantage of the floating-point representation used in computers. A floating-point number r_{ij} is represented by a sequence of bits using a mantissa/exponent format

$$r_{ij} = \pm m\, b^e \tag{7.43}$$

where the exponent e is an integer, the base b is also an integer, typically some power of two, and the mantissa m is a fraction normalized such that $1/b \le m < 1$. For example, in the 32-bit IEEE standard, $b = 2$, the first bit stores the sign information, the next eight bits hold the exponent e, and the remaining twenty-three bits represent the mantissa. Using shifting and masking functions, the exponent can be retrieved without any time consuming floating-point operations.

[6] The value k is often negative so a constant offset must be added to ensure the index to $N_\epsilon[\,]$ is positive. More on this point shortly.


```
begin find_cor_dim(x[][], N)
      set K = e_max − e_min + 1
      set offset = 1 − e_min
      set k_min = K
      set k_max = 1
      for k from 1 to K
          set N_ε[k] = 0
      endfor
      for i from 1 to N − 1
          for j from i + 1 to N
              set r = ‖ x[i][] − x[j][] ‖
              set e to the exponent of r
              set k = e + offset
              set N_ε[k] = N_ε[k] + 1
              if (k < k_max) then
                 set k_max = k
              endif
              if (k > k_min) then
                 set k_min = k
              endif
          endfor
      endfor
      set sum = 0
      for k from k_min to k_max do
          set sum = sum + N_ε[k]
          plot x = k versus y = ln(2 sum/N²)/ln 2
      endfor
end find_cor_dim
```

Figure 7.2: Pseudo-code for `find_cor_dim`.

To take advantage of this floating-point representation, choose $\epsilon_0 = 2^{e_{min}}$ where e_{min} is the smallest exponent possible in the floating-point representation, and choose

$$K = e_{max} - e_{min} + 1 \tag{7.44}$$

where e_{max} is the largest possible exponent. For each inter-point distance r_{ij}, increment $N_\epsilon[e - e_{min} + 1]$ by one where e is the exponent of the floating-point representation of r_{ij}. This exponent is offset by $e_{min} - 1$ to ensure that the index is positive.

Pseudo-code for the algorithm is presented in Fig. 7.2. and the output of the algorithm for Henon's map is shown in Fig. 7.3.

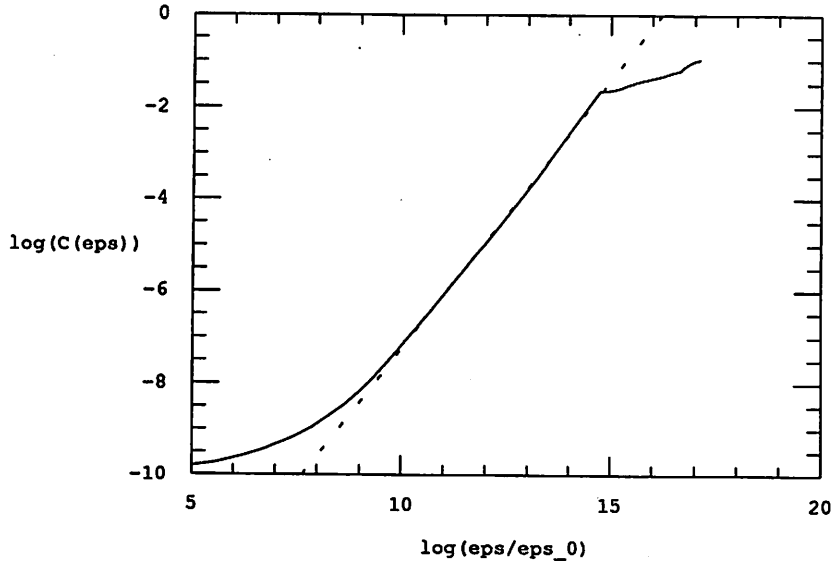


Figure 7.3: $\log_2 C(\epsilon)$ versus $\log_2 \epsilon$ for Henon's map: $x_{k+1} = y_k + 1 - 1.4x_k^2$, $y_{k+1} = 0.3x_k$. ϵ_0 is a constant that accounts for the biasing of the exponents—it does not affect the slope. The slope and, therefore, D_C is approximately 1.2. $N = 2000$ points were used in the calculation.

Remarks:

1. In the IEEE floating-point standard, the exponent is stored with a constant value, called a *bias*, added to it. The bias is chosen such that the exponent is never negative. Thus, $e_{min} = 0$ and $offset = 1$. When plotting, the slope is all that is sought and, therefore, the exponents (i.e., the k's) do not have to be debiased before plotting.

2. Owing to duplications, not all the inter-point distances are computed. Only the $(N^2 - N)/2$ values r_{ij} for $i = 1, \ldots, N-1$ and $j = i+1, \ldots, N$ are calculated. It follows that the entries in $N_\epsilon[\,]$ are half the actual count. This explains why *sum* is multiplied by two before plotting.

3. There will be some k_{min} and k_{max} such that $N_\epsilon[k] = 0$ for any $k > k_{max}$ and for any $k < k_{min}$. Only values of k from k_{min} to k_{max} are plotted.

4. If more points on the plot are desired, raise r to the mth power for some integer $m > 0$, and use $x = k/m$ when plotting. This

increases the number of points plotted by the factor m. A value of $m = 2$ is reasonable. It doubles the number of points plotted, and the cost is just one multiplication per r_{ij}.

kth nearest-neighbor dimension

If $g(k, D_{nn})$ were independent of k and D_{nn}, then the slope of a log-log plot of $\bar{r}(k)$ versus k would yield D_{nn}. $g(k, D_{nn})$, however, depends on k and D_{nn} and, therefore, a log-log plot of $\bar{r}(k)$ versus k is not a straight line. Fortunately, $g(k, D_{nn})$ can be approximated by (Pettis *et al.* [1979])

$$g(k, D_{nn}) \approx \frac{D_{nn} - 1}{2kD_{nn}^2} + \frac{(D_{nn}-1)(D_{nn}-2)}{12k^2 D_{nn}^3} - \frac{(D_{nn}-1)^2}{12k^3 D_{nn}^4} - \frac{(D_{nn}-1)(D_{nn}-2)(D_{nn}^2 + 3D_{nn} - 3)}{120k^4 D_{nn}^5}. \tag{7.45}$$

This expression leads to an iterative method for calculating D_{nn}. Pseudo-code for this algorithm is presented in Figs. 7.4 and 7.5.

Remarks:

1. The indexing scheme in `find_ln_r` is designed so that no inter-point distance is calculated twice.

2. The routine `calculate_d` (not shown) calculates the current iterate of d using the reciprocal of the standard least-squares formula

$$d = \frac{k_{max} \sum_{k=1}^{k_{max}} (\ln k)^2 - \left(\sum_{k=1}^{k_{max}} \ln k \right)^2}{k_{max} \sum_{k=1}^{k_{max}} \ln(k)\, g\_ln\_r[k] - \sum_{k=1}^{k_{max}} \ln k \sum_{k=1}^{k_{max}} g\_ln\_r[k]} \tag{7.46}$$

3. Equation (7.23) is valid only when the nearest-neighbor distances are small. It follows that the algorithm is accurate only for small values of k_{max}. What is meant by "small" depends on the number of data points. If N is increased, k_{max} can be increased too because a larger N implies that the

```
begin find_dnn(x[][], N)
      choose k_max, i_max, E_r, and E_a
      set ln_r[] = find_ln_r(k_max, x[][], N)
      set i = 0
      set d = 0
      repeat
            set i = i + 1
            if (k = k_max) then
               exit--no convergence
            endif
            set d_old = d
            for k from 1 to k_max do
                set g_ln_r[k] = ln_r[k] + g(k, d)
            endfor
            set d = calculate_d(k_max, g_ln_r[])
      until (|d - d_old| < d E_r + E_a)
      return (d)
end find_dnn

begin find_ln_r(k_max, x[][], N)
      for k from 1 to k_max do
          for i from 1 to N do
              set nn[i][k] = 0
          endfor
      endfor
      for i from 1 to N - 1 do
          for j from i + 1 to N do
              set r = ||x[i][] - x[j][]||
              call sort(r, k_max, nn[i][])
              call sort(r, k_max, nn[j][])
          endfor
      endfor
      for k from 1 to k_max do
          set ln_r[k] = 0
          for i from 1 to N do
              set ln_r[k] = ln_r[k] + nn[i][k]
          endfor
          set ln_r[k] = ln(ln_r[k]/N)
      endfor
end find_ln_r
```

Figure 7.4: Pseudo-code for `find_dnn` and `find_ln_r`.


```
begin sort(r, k_max, nni[])
      set k = k_max
      while (k > 0 and nni[k] = 0.0) do
            set k = k - 1
      endwhile
      while (k > 0 and r < nni[k]) do
            if (k < k_max) then
               set nni[k + 1] = nni[k]
            endif
            set k = k - 1
      endwhile
      if (k < k_max) then
            set nni[k + 1] = r
      endif
end sort
```

Figure 7.5: Pseudo-code for a bubble-sort routine `sort`. Called by `find_ln_r`.

nearest neighbors are closer. Somorjai [1986] suggests choosing $k_{max} = \alpha\sqrt{N}$ with $\alpha \approx 0.5$. This rule-of-thumb yields $k_{max} = 16$ for $N = 1000$.

Lyapunov dimension

Calculation of the Lyapunov dimension requires calculation of the Lyapunov exponents. See Section 3.4.3 for details.

Discussion

Calculation of distances All three of the dimension algorithms presented above require the calculation of inter-point distances. A single distance calculation is not costly, but given N points, there are roughly $N^2/2$ distances that need to be computed. Since $N = 10000$ is not uncommon, the distance calculations are the most time consuming part of the algorithms.

There are a few techniques that can be used to decrease the number of distances that are calculated. One approach is to choose a small set of N_{ref} random reference points $\{x_{i_{ref}}\} \subset \{x_i\}$. The distances from these reference points to the other x_i are used by the algorithm. This reduces the number of distance calculations to approximately $N_{ref}(N - N_{ref}/2)$ thereby decreasing the number of distance calculations by a factor of $2N_{ref}/N$.

```
begin ell_one_norm(n, r_max, x[])
      set r = 0.0
      for i from 1 to n
          set r = r + |x[i]|
          if (r > r_max) then
             return r_max
          endif
      endfor
      return r
end ell_one_norm
```

Figure 7.6: Pseudo-code for `ell_one_norm`.

A second technique that improves the efficiency of the algorithms is to use the ℓ_1 norm to calculate r,

$$r = \|u\|_1 := |u_1| + \cdots + |u_n| \tag{7.47}$$

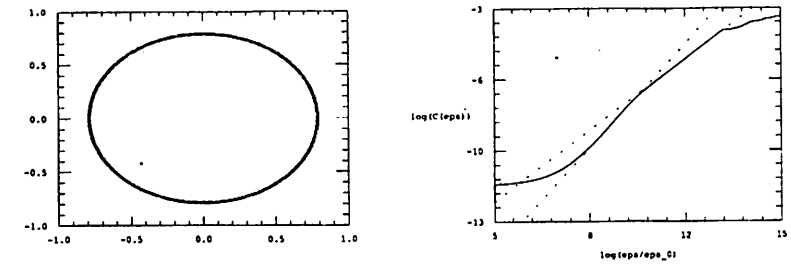
where $u = [u_1 \ \cdots \ u_n]^T$. The ℓ_1 norm avoids the multiplications and square root operation that are required to evaluate the Euclidean norm and reduces the number of floating-point operations by a factor of two.

A third technique is to calculate fully only those distances that are less than some limit r_{max}. The norm is usually calculated in a loop as is shown in Fig. 7.6. The distance calculation is aborted if d exceeds r_{max}. This approach can reduce the elapsed time for estimating the dimension by fifty percent. For D_{cap} and D_C, r_{max} is set to the largest value of ϵ that will be plotted. For D_{nn}, r_{max} is not constant; when r_{ij} is calculated, r_{max} is set to the maximum of the k_{max}th nearest neighbor of x_i and of x_j.

Calculation of the slope Typically, the slope of the log-log plot is determined using a least-squares fit,

$$m = \frac{K\sum_{i=1}^{K} x_i y_i - \sum_{i=1}^{K} x_i \sum_{i=1}^{K} y_i}{K\sum_{i=1}^{K} x_i^2 - \left(\sum_{i=1}^{K} x_i\right)^2} \tag{7.48}$$

where the points on the x-axis are $\{x_1, \ldots, x_K\}$ and the points on the y-axis are $\{y_1, \ldots, y_K\}$.



(a) (b)

Figure 7.7: A set with different dimensions at different length scales. (a) From a distance the set appears to be a circle; upon magnification, the set is seen to be an annulus; (b) a log-log plot of $C(\epsilon)$ versus ϵ.

Fig. 7.3 is typical of the log-log plots encountered when estimating dimension. It is clear from the figure that there is a limited range where the slope is approximately constant, and data only from this range should be used to estimate the slope. Unfortunately, there is currently no robust technique available to detect the useful portion of the plot automatically.

It can happen that the log-log plot has two or more regions with different slopes. Consider the set in Fig. 7.7 that, from a distance, appears to be a circle. Upon magnification, the set is seen to be a thin annulus, that is, it has non-zero width. At large length scales (i.e., for ϵ greater than the width of the annulus), the dimension of the set is 1. At smaller length scales (i.e., for ϵ less than the width of the annulus), the dimension is 2.

The dependence of the dimension on the length scale can be caused not only by the structure of the attractor, but by noisy data. Unless there is reason to believe otherwise, additive noise favors no particular direction in state space. Thus, additive noise tends to "fatten" the attractor. More precisely, if the attractor is embedded in $\mathbb{R}^n$, for length scales below the noise level, the dimension of the noisy attractor is n. As a simple example, let $x^* \in \mathbb{R}^2$ be a point on an attracting limit cycle. For some $\tau > 0$, let $\{x_i := \phi_{i\tau}(x^*)\}_{i=1}^N$ be the set of input points to a dimension-finding algorithm. Without noise, the x_i lie on a diffeomorphic copy of a circle, and the log-log plot has a single slope equal to 1. When noise is added to $\{x_i\}$, the noisy data lie in a thin annulus (as in Fig. 7.7(b)) and the log-log plot has two slopes. For ϵ greater than the noise level, the slope is 1. For ϵ less than the noise level, the slope is 2. See Ben-Mizrachi *et*

al. [1984] and Ott *et al.* [1985] for a further discussion of the effects of noise on the estimation of dimension.

As a final note of caution, we observe that the statistically rigorous derivation of kth nearest-neighbor dimension shows that the log-log plot of $\bar{r}(k)$ versus k is not a straight line—there is the $g(k, D_{nn})$ correction term. This result makes one wonder whether it is unrealistic to expect the log-log plots for the other types of dimension to be straight lines. Until this question is answered, numerical estimates of dimension should be interpreted carefully and any unexpected or unusual results should be corroborated by other methods.

Accuracy The statistics and accuracy of these algorithms are not well-understood. How many data points are needed for an accurate estimate of the dimension? What is the mean and variance of the estimate? How do the statistics depend on the dimension of the embedding space?

Interested readers are referred to Caswell and Yorke [1986], Holzfuss and Mayer-Kress [1986], and Somorjai [1986] for more discussion on this topic.

The curse of dimensionality The algorithms presented in this section are reasonably accurate for low-dimensional attractors embedded in low-dimensional spaces (i.e., $\mathbb{R}^n$ for $1 \leq n \leq 5$). When the embedding dimension n increases, the accuracy and efficiency of the algorithms decrease. This is due to the so-called "curse of dimensionality." As the embedding dimension is increased, more and more of the embedding space is empty. Thus, to achieve statistically reliable estimates, the number of data points must increase. Furthermore, in higher dimensions, distances tend to be distributed over a more narrow range. For instance, when d is large, points distributed uniformly in the interior of a d-dimensional hypersphere tend to lie near the surface of the hypersphere[7] and as $d \to \infty$, the standard deviation of the inter-point distances approaches 0. Thus, any dimension algorithm that relies on statistics over a range of inter-point distances encounters difficulty with high-dimensional spaces.

One approach that bypasses the curse of dimensionality is the method of projection pursuit. This technique uses low-dimensional projections to form an estimate of the probability density which is then used to calculate any of the probabilistic dimensions. Interested readers are referred to Friedman *et al.* [1984], Huber [1985], and Somorjai [1986].

[7] If r is a random variable uniformly distributed over the unit interval, then the random variable for the distance from the origin of points uniformly distributed in the interior of a d-dimensional unit hypersphere is $r^{1/d}$.

7.2 Reconstruction of attractors

In this section, we present a remarkable result, first proved by Takens [1980], that allows a strange attractor to be reconstructed from a sampled time waveform of just one component of the state. This is a useful technique in experimental settings. If data are gathered from measurements of a physical system, only one state variable needs to be measured, thereby cutting instrumentation and data storage costs. Moreover, for an infinite-order system or for a system where one or more of the state variables cannot be measured directly, reconstruction may be the only way to observe the attractor.

Let an attractor A of an nth-order system with flow ϕ_t be contained in an N-dimensional compact manifold $M \subset \mathbb{R}^n$. Define the reconstruction function $F: M \to \mathbb{R}^{2N+1}$ as

$$F(x) := [\phi_0^{(j)}(x) \;\; \phi_\tau^{(j)}(x) \;\; \cdots \;\; \phi_{2N\tau}^{(j)}(x)]^T \tag{7.49}$$

where $\phi_t^{(j)}(x)$ is the jth component of $\phi_t(x)$, j is arbitrary, and $\tau > 0$ is the sampling period, also arbitrary.

Generically, F is an embedding, that is, F diffeomorphically maps M onto some compact N-dimensional manifold $M' \subset \mathbb{R}^{2N+1}$.

This fact implies that given a sequence $\{y_k\} := \{\phi_{k\tau}^{(j)}(x)\}_{k=1}^K$ that corresponds to a uniformly time-sampled component of a trajectory that lies on an attractor A, the sequence of points

$$\begin{gathered}
[y_0 \;\; y_1 \;\; \cdots \;\; y_{2N}]^T \\
[y_1 \;\; y_2 \;\; \cdots \;\; y_{2N+1}]^T \\
\vdots \\
[y_i \;\; y_{i+1} \;\; \cdots \;\; y_{i+2N}]^T \\
\vdots \\
[y_{K-2N} \;\; y_{K-2N+1} \;\; \cdots \;\; y_K]^T
\end{gathered} \tag{7.50}$$

lies on a diffeomorphic copy of A.

Several examples of reconstructed attractors are presented in Fig. 7.8. The reconstructed versions are definitely different from