Fast Multifractal Analysis by Recursive Box Covering

B. F. Feeny

Department of Mechanical Engineering, Michigan State University,
East Lansing, Michigan 48824-1226 U.S.A.

Abstract

A fractal can be generated by using recursion, and fractal set of data can be analyzed in the same way. This paper regards a box-counting algorithm which exploits self-similarity via recursion. The speed of the algorithm is $O(n)$, where n is the number of points in the set. The algorithm is applied to multifractal analysis of the binomial multiplicative process. The algorithm does not involve sorting the floating-point summation; a brief examination of the sorting issue is presented. Scattering for negative q, characteristic of box-counting algorithms, is encountered, and is explained for the simple case of a uniform 1-D set.

1. Introduction

The computation of fractal dimensions is important in many fields, such as chaotic dynamics, turbulence, geology, physiology, and satellite imagery. There are many different ways of computing dimensions. Commonly used dimensions are the fractal dimension (limit capacity), information dimension, and correlation dimension. In monofractal analysis, the analyst chooses one of these in the dimensionality study. Chaos analysts have typically chosen correlation dimension or information dimension for efficiency [Moon, 1992]. In general, however, different types of dimension calculations will sometimes produce different values for the dimension of the set. In such case, the set constitutes a multifractal, and can actually be characterized by a spectrum of dimensions.

Box counting is one method for computing generalized dimensions in point sets. Some memory-efficient box-counting algorithms involve data sorting [Barth *et al.*, 1992; Block *et al.*, 1990; Meisel *et al.*, 1992; Hou *et al.*, 1990; Liebovitch and Toth, 1989]. The top speed of such a code is $O(n \log n)$, where n is the number of data. The agglomeration method [Meisel *et al.*, 1992], on the other hand, is not memory efficient, but its speed is independent of n. Chachere [1992], Grassberger [1993], and Molteno [1993] reported $O(n)$, memory-efficient algorithms for box counting.

In this paper, we discuss an $O(n)$ box-counting algorithm, and apply it to multifractal analysis. The method performs no sorting. It is based on recursion. It consists of a function which zooms into a subset of the data region, and identifies the data which lie therein. Then, as if this subregion were an original region, the function calls itself, zooms and identifies. This process continues. Thus, the algorithm itself is self-similar.

The algorithm is similar to that of Molteno [1993]. This work differs by pursuing a broad range of generalized dimensions, including the estimations of the singularity spectrum $f(\alpha)$ (to be discussed shortly), varying the resolution in the box sizes, and addressing the effect of not sorting the partition sum.

This paper is organized as follows. The next section summarizes the theoretical de-

velopment of multifractals and the application of box-counting. This is followed by a description of the fast algorithm for box-counting multifractal analysis. The absense of sorting is then discussed The algorithm is then applied, and a discussion on scattering is included.

2. Theory

Fractals have been defined as objects which are self similar at various scales, or sets which have fractional dimension [Feder, 1988].

We start by discussing dimension in a box-covering sense. The fractal dimension, or limit capacity, of a set is defined as the scaling exponent d_0 in the relationship

$$N(r) \sim Ar^{-d_0}, \qquad r \to 0, \tag{1}$$

where $N(r)$ is the number of boxes of length, or "size", r necessary to cover the set, and A is a constant [Mandelbrot, 1983; Feder, 1988].

Suppose the set is represented by a large number of points. If these points are uniformly distributed across the fractal, then the fractal dimension completely characterizes the dimension of the set. However, the situation gets more interesting if the points are not distributed uniformly. It is possible that the mass distribution of the points varies. If so, at a given box length r, it is possible to identify regions of the same masses μ. In a *multifractal*, these regions of mass μ comprise a fractal subset of the original set [Feder, 1988]. Introducing the scaling exponent α, such that

$$\mu = r^{\alpha},$$

the dimension of mass distributions can be plotted as a function $f(\alpha)$. This is called the dimension spectrum.

The spectrum $f(\alpha)$ is not always computed directly. People often choose to compute another representation of generalized dimensions. These can be described through *mass exponents*. The mass, or probability density, can be estimated within a box of size r as $\mu_i = n_i/n$, where n_i is the number of points in the box, and n is the total number of points. Then a measure can be constructed as

$$M_d(q,r) = \sum_{i=1}^{N} \mu_i^q r^d,$$

where N is the number of boxes that cover the set. As $r \to 0$, $M_d(q,r) \to 0$ if $d > \tau(q)$, and $M_d(q,r) \to \infty$ if $d < \tau(q)$. $d = \tau(q)$ is called the mass exponent, at which $M_d(q,r)$ converges to some finite value. It is customary to define $Z(q,r) = \sum_{i=1}^{N} \mu_i^q$ as the *partition function*. Then $Z(q,r) \sim r^{-\tau(q)}$, and thus,

$$\tau(q) = \lim_{r \to 0} \frac{\log Z(q,r)}{\log r}.$$

From this, one can define generalized dimensions called Rényi dimensions, defined as [Rényi, 1970; Grassberger, 1983; Hentschel and Procaccia, 1983]

$$D(q) = \frac{\tau(q)}{(q-1)} = \lim_{r \to 0} \frac{\log Z(q,r)}{(q-1)\log r}.$$

In the limit as $q \to 1$, this expression reduces to

$$D(1) = \lim_{r \to 0} \frac{\sum_{i=1}^{N} \mu_i \log \mu_i}{\log r}.$$

$D(1)$ is the information dimension.

The dimension functions $D(q)$ and $f(\alpha)$ are related through a Legendre transformation [Halsey *et al.*, 1986]:

$$\alpha(q) = -\frac{d}{dq}\tau(q)$$

$$f(\alpha(q)) = q\alpha(q) + \tau(q).$$

Typically, the $D(q)$ are computed by estimating $Z(q,r) = \sum \mu_i^q$ for various values of r, and then by fitting the slope in the plot of $\log Z(q,r)/(q-1)$ vs. $\log r$.

One way to compute the μ_i is to cover the set with boxes of size r. In box i there are n_i boxes. The probability associated with each box is $\mu_i = n_i/n$. The partition function is then $Z(q,r) = \sum_{i=1}^{n_r} \mu_i^q$, where n_r is the number of boxes of size r that cover the set. After doing this for many values of r, $D(q)$ is approximated from the slope of the plot of $\log Z(q,r)/(q-1)$ vs. $\log r$.

Box counting has some drawbacks, such as scattering for negative q [Borgani *et al.*, 1993; Meisel *et al.*, 1992; Aharnoy, 1990; Buczkowski *et al.*, 1988], and other sources of error [Barth *et al.*, 1992]. Methods for reducing scattering and other errors have been proposed [e.g. Pastor-Satorras and Riedi, 1996; Yamaguti and Prado, 1997; Alber and Peinke, 1998]. Alternatives to this fixed-volume box counting include fixed-mass box counting [Badii and Broggi, 1988; Mach *et al.*, 1995], generalized correlation integrals [Pawelzik and Schuster, 1987], and direct computation of the $f(\alpha)$ spectrum [Chhabra and Jensen, 1989; Meneveau and Sreenivasan, 1989; Yamaguti and Prado, 1995]. The use of wavelet transforms [Muzy *et al.*, 1994] is a powerful method free of scattering problems, and employs a broadly used transform tool. There are further options if the data is generated by a dynamical process. One such method uses the additional information of the dynamics of the set, and estimates probabilities by looking at return times in the iterations of points. This approach produced very accurate results for the circle map and Bénard convection data [Jensen *et al.*, 1985]. Another is based on the extraction of unstable periodic orbits [Grebogi *et al.*, 1988].

Despite its drawbacks, box counting continues to be used [recent examples include Kurokawa *et al.*, 1999; Laferriere and Gaonach, 1999; Fernandez *et al.*, 1999], perhaps because it is directly related to the presentation of multifractal theory through partition functions, making it a natural way to approach multifractal computations. An any case, this paper focuses on box counting. The task is to describe an efficient algorithm for computing the μ_i in the cover of boxes with size r. Our focus is on estimating $Z(q,r)$ and

hence generating $D(q)$. We choose the Legendre transform for converting to $f(\alpha)$. The usage of the Legendre transformation and its associated issues are not addressed here, but can be found elsewhere [e.g. Chhabra and Jensen, 1989].

3. Self-Similar Box-Counting Algorithm

The basic idea applied in this algorithm is to use a recursive function, i.e. a function that calls itself. This gives rise to an algorithmic structure that is self similar. This is essentially the same structure as that presented by Molteno [1993]. Here, we apply it in the computation of the partition functions.

We want to build the partition functions $Z(q,r)$ for a set S of points in some E-dimensional embedding space. Let us first think of a set embedded in two dimensions. The coordinates of the data are stored in a $n \times 2$ array called DATA. The set of points can be normalized to lie in the unit square $I \times I$, where $I = [0,1]$. The axes of the set can be stretched proportionately to avoid dimensional distortions. We will cover the set with square boxes whose side length is r. Trivially, one box of size $r = 1$ will cover the set S. This corresponds to the zeroth order of the fractal.

The box-covering method requires that we shrink the box size r. To produce evenly distributed results, this should be done in logarithmic increments. Suppose the box size diminishes by a factor of two. The zeroth order covering can now be subdivided into four boxes. These four subregions represent the first-order level of the fractal algorithm. By checking each box to see if any of the points of S are included, we can see which boxes are part of the first-order covering. Each subregion which contains points in S is then viewed as its own original zeroth-order region, and is subdivided into four smaller boxes.

Figure 1 shows four boxes in the first level of the algorithm. The boxes are labeled (0,0), (0,1), (1,0), and (1,1). Together they cover the image. Suppose some subset of the data lies in box (0,0). These m_1 points are inventoried by an integer array INDEX of length m_1. This array includes the row indices of the original data set DATA corresponding to those points which lie in (0,0) of the first level. The length scale shrinks by a factor of two, and the box is subdivided into four smaller boxes. These smaller boxes, part of the second level of the algorithm, are again labeled (0,0), (0,1), (1,0), and (1,1). One by one, the program checks which, if any, of the m_1 points are contained in the subregion. If so, this subregion is part of the second-level cover. Suppose there are m_2 points in this subregion. (Note that each subregion will have a different number of points, and thus a different value for m_2.) Then, a global partition function, whose index corresponds to the level of magnification, is incremented by $(m_2/n)^q$. (Note that when $q = 0$, this reduces to simple box-covering, as in a limit-capacity calculation.) The program then zooms into the subregion, taking with it an integer array of length m_2 which contains the row indices of DATA included in the subregion. At this new level of magnification, the process is repeated.

To prevent this from being an infinite recursion, the zooming function only calls itself if a certain maximum level of magnification has not been reached.

The result is a logarithmically spaced output array of box sizes, and an output array of partition functions for each box size, which reduces to the number of boxes in the cover for

$q = 0$. Since the spacing of the box sizes is logarithmic, it does not take many increments to span the meaningful range of box sizes. To increase the output resolution, the entire process is repeated with increments in the initial, zeroth-order region. These increments are made logarithmically between one and two. For example, to increase the resolution by a factor of ten, the initial box size is incremented by lengths $2^{n/10}$, $n = 1, \ldots, 10$. An outline of the algorithm for two-dimensional sets is as follows:

0. Read and normalize the data such that it lies in the unit square $I \times I$. Array DATA contains n points, array INDEX contains n indices, and LEVEL is initialized at the current level of zero. Initialize the zeroeth-order region $R_o = I \times I$.
1. Subdivide this region into four equal subregions by cutting the length scale by a factor of two.
2. Using subrouting ZOOM, zoom into each subregion, taking with it the current LEVEL and the array INDEX of the indices of all data found in the mother region. At level zero, this means all indices. Increment LEVEL = LEVEL + 1. The subregion is a box of size $r = 2^{-LEVEL}$.
3. In the current subregion, identify which of the indices in INDEX are associated with data in that subregion, and store them in newINDEX, defined within the subroutine. The length n_{new} of newINDEX indicates the number of points in the subregion or box, and its probability measure is $p = n_{new}/n$. Increment the partition function $Z(q, r) = Z(q, r) + p^q$.
4. If a maximum zooming level MAXLEVEL has not been reached, and if newINDEX is not empty, repeat step 2, with newINDEX playing the rôle of INDEX.
5. Repeat step 1. for a logarithmically incremented initial region R_o. This will increase resolution in the $\log r$ output.

At each level, n indices are checked, and n components of the partition sum are performed. Hence, there are an order of $n*$MAXLEVEL computations. Therefore the speed of the algorithm is $O(n)$.

Let us consider the memory requirements. We require a data array of length $n \times 2$. This is a global variable. There is also an array INDEX which has a maximum length of n. A new INDEX is created locally in each subrouting. At any given instant, the program is in, at most, MAXLEVEL subroutines, each of which defines INDEX locally. Thus, together with DATA, there is a storage requirement of n*(MAXLEVEL + 2), plus the contribution of a few extraneous variables. Thus, the memory requirement is also $O(n)$.

4. No Sorting

This algorithm does not perform sorting. Sorting is generally required for computer summations to prevent the contribution of small numbers from being lost when added to numbers of much greater magnitude. In the case of fractal partition functions, the concern is that extremely large values of μ_i^q may overshadow smaller values of μ_i^q such that the latter are lost when added together in the floating point sum. This would be detrimental if the small values of μ_i^q, by their distribution in the fractal set, have and important contribution to the true sum. However, since sorting is an $n \log n$ operation, it would defeat the speed advantage of this algorithm. In any case, this algorithm is applied in a later section, and

good results are obtained. Grassberger [1993] has also proposed an $O(n)$ algorithm, and states outright that no sorting is performed in the calculation.

Do multifractal box counts require sorting? If so, under what conditions? It is likely that such questions can be answered by examining the sum and applying the properties of multifractals. The ideas in this section are intended to gain some insight as to whether sorting may be significant, and, if so, under what conditions.

The partition sum involves $\sum_{i=1}^{N} \mu_i^q$, where $\mu_i = r^{\alpha_i}$, and the subscript on α_i is defined to correlate a value of α with a mass μ_i found in a box of size r. When $q > 0$, the maximum true quantity in the partition sum is $[\mu^q]_{max} = r^{\alpha_{min}} q$, and the minimum true quantity in the partition sum is $[\mu^q]_{min} = r^{\alpha_{max}} q$. Likewise, when $q < 0$, the maximum true quantity in the partition sum is $[\mu^q]_{max} = r^{\alpha_{max}} q$, and the minimum true quantity in the partition sum is $[\mu^q]_{min} = r^{\alpha_{min}} q$. The reference to a "true" quantity is in regard to the expected occurrence of anomolous densities found in boxes which are partially filled, due to either finite data limitations or partial overlap of the box onto the substrate of the data.

Let us reformulate the sum in terms of a discretized α space, as done, for example, by Dubrulle and Lachièze-Rey [1994] and also in integral form by Halsey *et al.* [1986]. If the α space were discretized into M small intervals, each interval would be associated with a number N_j of associated boxes, defined by the multifractal distribution, such that $N_j = Ar^{-f(\alpha_j)}$, as $f(\alpha_j)$ is the dimension of the approximate distribution of masses $\mu_j = r^{\alpha_j}$. N_j is approximate in the sense of a discrete α interval, a finite r, and a finite data set. Then, we could approximate the partition sum in terms of this sum on the α intervals, such that $Z(q,r) = \sum_{j=1}^{M} N_j \mu_j^q = \sum_{j=1}^{M} A r^{\alpha_j q - f(\alpha_j)}$, where $\alpha_1 = \alpha_{min}$ and $\alpha_M = \alpha_{max}$.

The values in this sum are expected to vary widely in magnitude, perhaps enough to warrant a sorted sum. Let us suppose that the sum has a dominant term, or at least a maximum term, corresponding to a dominant value of α. This maximum term occurs where

$$\frac{d}{d\alpha}(r^{\alpha_j q - f(\alpha_j)}) = (q - f'(\alpha)) r^{\alpha_j q - f(\alpha_j)} \ln(r) = 0,$$

i.e. the maximum thus occurs when $q = f'(\alpha)$. This relationship is in essence the Legendre transformation relating $f(\alpha)$ to $\tau(q)$ [Halsey *et al.*, 1986; Dubrulle and Lachièze-Rey, 1994]. Under this condition,

$$\frac{d^2}{d\alpha^2}(r^{\alpha_j q - f(\alpha_j)}) = -f''(\alpha) r^{\alpha_j q - f(\alpha_j)} \ln(r) < 0$$

since $f''(\alpha) < 0$, confirming a local maximum [Halsey *et al.*, 1986]. The minimum term occurs at one of the endpoints in the sum. Thus the dominant term in the partition sum corresponds to the contribution of masses of dimension $f(\alpha)$, or those distributed according to $D(q)$, and does not correspond to terms of maximum μ^q.

For example, when $q = 0$, the dominant or maximum term of the partition sum occurs at the value of α where $f(\alpha)$ is maximum, i.e. where $f(\alpha) = D(0)$. (In such case, all values of $\mu^q = 1$, and so no sorting is needed in computing $D(0)$.) When $q > 0$ is very large, the highest term in the sum is given by α near α_{min}, where μ_j^q is nearly its maximum. The μ_j^q dominating terms are "similar" in magnitude to the maximum values. Terms lost in the sum by not sorting, should such terms exist, are distant on the α axis, and are far

from dominating. Hence, it is conceivable that the terms that may be lost by not sorting may be terms that are not significant to the entire sum. Similarly, if $q < 0$ is large in magnitude, the largest contributing μ_j^q correspond to a value of α near α_{max}, and thus the largest contributing μ_j^q are near $[\mu^q]_{max}$. Again, it is conceivable that the terms lost by not sorting may not be significant to the partition sum.

If we consider small values of q, we find that the largest contributing terms in the sum correspond to values of α which produce relatively high values of $f(\alpha)$. Thus, it is the large numbers of boxes with the corresponding masses that contribute most significantly, rather than fewer numbers of extreme masses. The hope is that the values of μ_j^q of the maximum contribution to the sum do not differ from the $[\mu^q]_{max}$ to the extent that $\mu_j^q + [\mu^q]_{max} = [\mu^q]_{max}$ in a floating-point sum, i.e. that the maximum value does not eclipse the contributing values when summoned in the sum.

To this end, let us consider the floating point sum $x + y = x$, or $1 + y/x = 1$, due to an extremely small ratio of $\epsilon = y/x$. The machine epsilon ϵ can be estimated in a while loop, in which ϵ starts with the value 1.0, and is successively halved until $1 + \epsilon = 1$ is satisfied in the machine [Forsythe *et al.*, 1977]. Running this loop with Matlab on our current machine, the equation is satisfied when $\epsilon = 2^{-\Delta}$, with $\Delta = 54$.

In our problem, the sum of terms $[\mu^q]_{max}$ and μ_j^q would be "meaningful" if $[\mu^q]_{max}/\mu_j^q = r^{\alpha_{min} q} - r^{\alpha_j q} < 2^{-\Delta}$ (for $q > 0$). This requirement reduces to

$$(\alpha_j - \alpha_{min}) > -\Delta/q \log_2(r).$$

Suppose $\Delta = 52$ (for a couple of digits of substance), $\alpha_j - \alpha_{min} = 1$, and our scaling region runs through $\log_2(r) = -5$. Then we may be satisfied with q up to values of ten. A likewise discussion can be made for negative q.

Thus, depending on parameters such as the range of α, it may turn out that sorting is not crucial for smaller magnitudes of q since the values of μ_j^q do not differ excessively. For larger ranges of α (or $D(q)$), and smaller the scaling ranges r, this statement weakens. Parameters in some systems may be such that the structure of the partition sum with values of q of intermediate magnitude is not as clear. More rigorous studies could be made into this, if there is interest in performing $O(n)$ box counting computations. A quantitative estimate of the error as a function of the scaling exponents, machine epsilon, and box scale, would then be beneficial.

5. Application and Discussion

The speed of the program has been investigated with uniform two-dimensional data, and one-dimensional data embedded in $I \times I$. The two-dimensional data requires recursion at all levels in all four quadrants of box-reduction scheme, and hence the time requirement should roughly provide an upperbound on the time needed for sets embedded in two dimensions. The one-dimensional data should provide a lower bound for sets of dimension greater than one embedded in two dimensions. Thus, the usage of uniform data for depicting the speed should not accompany any loss of generality. The output resolution was one box count per binary decade in r. The program ran on a Sun 4/690 equipped with 4

central processing units, under a load of 20-25 users. Figure 2 displays a plot of cpu vs. n (in thousands), indicating that the algorithm speed is $O(n)$. The similarity in speeds suggests that n has a stronger influence on speed than the distribution of the data.

Because of this order of speed, there will be some value of n where it becomes more efficient than an algorithm of speed $O(n \log n)$. Molteno [1993] provides a quantititive comparison between the speeds of recursive box covering and the sort-based method of Hou *et al.* [1990].

The program's accuracy has been tested on 16K samples of a stochastic binomial multiplicative process [Feder, 1988]. We do not investigate the idiosyncrasies of box counting (through examing many sets), since the nature of box counting has been studied in other works. For any box-counting multifractal analysis, the results should be closely scrutinized before they are trusted.

The binomial multiplicative process was generated by using an iterated function system [Barnsley, 1988]. The functions were $f_1(x) = x/2$, and $f_2(x) = 1/2 + x/2$. The probabilities associated with the two functions were skewed, with $p_1 = 0.25$ and $p_2 = 0.75$, leading to a multifractal distribution of points. Figure 3 depicts the results for $D(q)$ and $f(\alpha)$, with a box-counting computation performed only at box sizes $r = 2^{-i}$, where i is an integer. The circles indicate calculated values, and the solid lines show the theoretical curves. In reference to the discussion on sorting in Section 4., the values of $\alpha(q)$ in Figure 3 are rather near the extremes for, say, $|q| = 5$, and the scaling region runs through, say, $\log_2(r) = -6$ (Figure 4). Thus, we might not expect significant sorting problems throughout the computation.

A comparison of $\log Z(q,r)/(q-1)$ vs. $\log r$ for $q = -5$ at two different output resolutions is shown in Figures 4 and 5. Figure 4 displays the plot for box sizes of 2^{-i}, and Figure 5 shows the scattering seen at ten times the resolution of Figure 4.

It turns out that, for box sizes of 2^{-i}, the structure of the box-counting algorithm perfectly matches that of the binomial multiplicative process, which is also defined in terms of powers 2^{-i}. For this reason, the results are especially clean. Generally, sets do not have the same internal structure as the box-counting algorithm. This invariably leads to scattering problems in the log-log plots for $q < 0$. Since the Legendre transformation from $D(q)$ to $f(\alpha)$ involves differentiation, errors from scattering are magnified. Thus the present example allows us to witness the performance of the summing algorithm independent of box-counting scattering issues.

5.1 On scattering

An oft-cited cause of scattering is the occurrence of partially "filled" boxes in the partition function [e.g. Yagamuti and Prado, 1997; Chhabra and Jensen, 1989; Borgani *et al.*, 1993]. If there are uncharacteristically thinly filled boxes, the associated μ_i will be uncharacteristic to the fractal. For example, suppose we have 1000 points covered by 10 boxes. If each box contains 100 points, then the partition function will have a value of $N(q,r) = 10 * 0.1^q$. For $q = -5$, this gives $N(-5,r) = 10^6$. Instead, suppose there are 1001 points covered by 11 boxes, 10 of which contain 100 points, and one which contains one point. Then $N(q,r) = 10 * (100/1001)^q + (1/1001)^q$. For $q = -5$, the term $(1/1001)^q$ dominates, leading to $N(q,r) \approx 10^{15}$. Thus, an outlying point or an edge point can lead

to large variations in $N(q,r)$.

To test this hypothesis, we have calculated the box-counting measures for a uniform unit interval I. The covering geometry in this example is as follows. Boxes (or measuring sticks) of length r are stacked, starting at the origin (the left endpoint of I), along the interval so as to cover I. The $\log_2 r$ resolution is chosen to be ten samples per decade (where a decade is an interval of $\log_2 2 = 1$). We can calculate the partition function for the covering of the interval. For ten uniform increments in $\log_2 r$, the starting box has length $2^{j/10} - l$, where $j = 1, 2, \ldots, 10$, and l is the level l of magnification. The cover will consist of boxes which are completely filled, and a partially filled box near the right endpoint of I. The measure in this partially filled box is $\mu_p = 1 \bmod r$. The measure in the full boxes is $\mu = r$. The number of full boxes in the cover of I is $M = (1 - \mu_p)/r$. Thus, the resulting partition function is $Z_I(q,r) = Mr^q + \mu_p^q$. Figure 6 shows plot of $\log_2 Z_I(-5,r)$ vs. $\log_2 r$ for this uniform interval as compared to $Z(-5,r)$ computed by the box-counting code for $n = 16K$ random data on the unit interval. Here, the box-counting code uses the same covering geometry as used in the "theoretical" calculation of $Z_I(q,r)$. The theoretical calculation, given by + symbols, represents the limiting case as $n \to \infty$. The plot for $n = 16K$ points is given by the o symbols. The similarity indicates that partial coverings on the boundaries of sets causes scattering. Since the geometry of this set is "clean", the scattering has a well-defined pattern for negative q. More arbitrary sets would probably produce less-discernable scattering patterns.

For the sake of illustration, we omit the contributions of boxes near the edges of the set by using *a priori* information, and thereby reduce scattering. In the $f(\alpha)$ curve, $\alpha_{min} = D(\infty)$ and $\alpha_{max} = D(-\infty)$, and $\alpha_{min} < \alpha < \alpha_{max}$, with $\mu_i = r^{\alpha_i}$. If the data represents a well-ordered multifractal, there should primarily be contributions to probability measures in the range $r^{\alpha_{max}} \leq \mu \leq r^{\alpha_{min}}$. Knowing α_{max}, we know the lower bound on the range of values for μ, and thus omit the contribution of any box with μ below this range. While this is an "engineering" remedy that needs *a priori* information, it provides an illustration of the computation with respect to scattering and resolution. Indeed, in some cases, such insight might come from a scattered, unmodified calculation, for example from a plot such as that shown in Figure 5. If scattering is due to partially filled boxes, as discussed above, then the scattered computations would contribute data in the plot below the "true" log-log curve.

Figure 7 shows partition functions $Z(-5,r)$, for the binomial process, calculated by omitting any $p < r^{\alpha_{max}}$, with $\alpha_{max} = 2$. This figure can be compared to Figure 5, which was computed with no modifications. Figure 8 shows the resulting $D(q)$ and $f(\alpha)$ curves, plotted with the symbol o, as compared to the theoretical solid curves. Without modifications, the $f(\alpha)$ curve estimated from computed outputs with high resolution in r suggests an $\alpha_{max} \approx 1.8$, compared to a theoretical $\alpha_{max} = 2$.

More rigorous approaches to scattering can be found [e.g. Borgani *et al.*, 1993, Pastor-Satorras and Riedi, 1996; Yamaguti and Prado, 1997].

Typically, however, box counting provides good estimates of $D(q)$ for positive q. Thus, it is acceptable for the commonly computed quantities $D(0), D(1)$, and $D(2)$.

Noting that this code rigidly adheres to a binary subdividing structure, it might be fruitful to consider other possiblities. Barnsley [1988] suggested that many fractal images

can be easily approximated by iterated function systems (IFS). The iterated functions could be identified by taking advantage of the "collage theorem". It seems reasonable that, if the resulting collage consists of non-overlapping pieces, then the IFS could be exploited in the box-counting code. The trick would be make sure that each point does not get counted into more than one box (in case the collage overlaps), and that the box sizes used in the cover are uniform. If this works for some examples, it seems that the structure of the code might agree with the structure of the fractal, and scattering might be reduced.

6. Higher Dimensions

It is often desireable to calculate dimensions for sets with an arbitrarily large embedding dimension E. This is especially the case when analyzing an experimental time series, in which one tries to reconstruct the strange attractor from time-series data [Takens, 1981; Broomhead and King, 1986; Gershenfeld, 1988]. Either E is unknown, and the dimensionality study is conducted through many values of E, or E is predetermined via a "nearest-neighbor" [Kennel *et al.*, 1992] or singular-systems [Broomhead and King, 1986] study.

The above box-counting algorithm zooms into subregions one by one. For squares in the plane, it does this by two nested loops. In E dimensions, this requires E nested loops. Since there are E nested loops of binary subdivision, the speed of the algorithm will be $O(2^E)$. Indeed such a trend is observed in numerical examples. Thus, the algorithm may not be efficient for high-dimensional data.

7. Conclusion

We have discussed an algorithm for box counting that is $O(n)$ in both speed and memory requirement. The algorithm, similar to Molteno's [1993], is based on recursive subdivisions of the covering grid. The speed is similar to the algorithms previously presented by Chachere [1992] and Grassberger [1993], reinforcing that box counting can be efficient.

This algorithm has been applied to multifractal analysis of data generated by the binomial multiplicative process. Computed results were compared to theoretical results. The algorithm does not perform sorting. We looked at the dominant terms in the partition sum, and found that when q is large in magnitude, the dominant terms are among the largest magnitude terms in the sum, and hence unsorted sums should not lose the dominant terms. For q that are low in magnitude, the span of terms in the partition sum, μ^q, are less likely to span the limits of machine epsilon. In any case, there is no concern about sorting when computing $D(0)$, and little concern with the other common quantities $D(1)$ and $D(2)$.

Inherent to box-counting algorithms, there is a problem of scattering in the partition functions $Z(q,r)$ for negative values of q. One cause of scattering is the low-probability associated with points in boxes near the edge of a fractal set. For illustration, we used a coarse modification in which, essentially, the contributions of these boxes are omitted. The

A priori information needed for this modification may not be available in all applications, but in some cases such information may be ascertained from a preliminary scattered, unmodified computation.

While the algorithm was described in two-dimensional embedding space, it can be extended to higher dimensions. In terms of the embedding dimension, the speed is $O(2^E)$. Thus, for large E, such as in the reconstructed phase space of a dynamical system, the algorithm may lose its advantage.

Acknowledgement

I thank Jay Treacy, who once showed me how to write a recursive C program for generating the Sierpenski gasket.

References

Aharnoy, A. [1990] 'Multifractals in physics: successes, dangers and challenges,' *Physica A* **168**, 479-489.

Alber, M., and Peinke, J. [1998] 'Improved multifractal box-counting algorithm, virtual phase transitions, and negative dimensions,' *Physical Review E* **57**(5) 5489-5493.

Badii, R., and G. Broggi [1988] 'Measurement of the dimension spectrum $f(\alpha)$: fixed-mass approach,' *Physics Letters A* **131**(6), 339-343.

Barnsley, M. [1988] *Fractals Everywhere* (Academic Press, New York).

Barth, A., G. Baumann, and T. F. Nonnenmacher [1992] 'Measuring Rényi dimensions by a modified box algorithm,' *Journal of Physics A* **25**, 381-391.

Block, A., W. von Bloh, and H. J. Schellnhuber [1990] 'Efficient box-counting determination of generalized fractal dimensions,' *Physical Review A* **42**(4), 1869-1874.

Broomhead, D. S., and King, G. P. [1986] 'Extracting qualitative dynamics from experimental data,' *Physica D* **20**, 217-236.

Borgani, S., G. Murante, A. Provenzale, and R. Valdarnini [1993] 'Multifractal analysis of the galaxy distribution: reliability of results from finite data sets,' *Physical Review E* **47**(6), 3879.

Buczkowski, S., Hildgen, P., and Cartilier, L. [1998] 'Measurements of fractal dimension by box-counting: a critical analysis of data scatter,' *Physica A* **252** 23-34.

Chachere, G. [1992] 'A fast O(N) and memory-efficient algorithm for box counting,' abstracts of the *SIAM Conference on Applications of Dynamical Systems,* Snowbird, Utah, October 15-19.

Chhabra, A., and Jensen, R. V. [1989], 'Direct determination of the $f(\alpha)$ singularity spectrum,' *Physical Review Letters* **62**, 1327-1330.

Dubrulle, B., and Lachièze-Rey, M. [1994] "On the multifractal analysis of galaxy catalogs with box-counting methods," *Astronomy and Astrophysics* **289**, 667-672.

Feder, J. [1988] *Fractals* (Plenum Press, New York).

Fernandez, E., Bolea, J. A., Ortega, G., and Louis, E. [1999] 'Are neurons multifractals?' *Journal of Neuroscience Methods* **89**(2), 151-157.

Forsythe, G. E., Malcolm, M. A., and Moler, C. B., [1977] *Computer Methods for Mathematical Computations* (Prentice-Hall, Englewood Cliffs).

Gershenfeld, N. [1988] 'An experimentalist's introduction to the observation of dynamical systems,' in Hao B.-L. (ed.), *Directions in Chaos, II* (World Scientific, Singapore), pp. 310-384.

Grassberger, P. [1983] 'Generalized dimensions of strange attractors,' *Physics Letters A* **97**(6), 227-230.

Grassberger, P. [1993] 'On efficient box-counting algorithms,' *International Journal of Modern Physics C,* **4**(3) 515-523.

Grebogi, C., Ott, E., and Yorke, J. [1988] 'Unstable periodic orbits and the dimensions of multifractal chaotic attractors,' *Physical Review A* **37**(5) 1711-1724.

Halsey, T. C., Jensen, M. H., Kadanoff, L. P., Procaccia, I., Shraiman, B. [1986] 'Fractal measures and their singularities: The characterization of strange sets,' *Physical Review A* **33**(2) 1141-1151.

Hentschel, H. G. E., and I. Procaccia [1983] 'The infinite number of generalized dimensions of fractals and strange attractors,' *Physica D* **8**, 435-444.

Hou, X.-J., R. Gilmore, G. Mindlin, and H. Solari [1990] 'An efficient algorithm for fast $O(N * \ln(N))$ box counting,' *Physics Letters A* **151**(1,2), 43-46

Jensen, M. H., L. P. Kadanoff, A. Libchaber, I. Procaccia, and J. Stavans [1985] 'Global Universality at the onset of chaos: results of a forced Rayleigh-Bénard Experiment,' *Physical Review Letters* **55**(25), 2798-2801.

Kennel, M. B., R. Brown, and H. D. I. Abarbonel [1992], 'Determining embedding dimension for phase-space reconstruction using a geometrical construction,' *Physics Review A* **45**, 3403-3411.

Kurokawa, T., Morikawa, M., and Mouri, H. [1999] 'Scaling analysis of galaxy distribution in the CfA2 data,' *Astronomy and Astrophysics* **344**(1), 1-6.

Laferriere, A., and Gaonach, H. [1999] 'Multifractal properties of visible reflectance fields from basaltic volcanoes,' *Journal of Geophysical Research—Solid Earth* **104**(B3), 5115-5126.

Liebovitch, L. S., and T. Toth [1989] 'A fast algorithm to determine fractal dimensions by box counting,' *Physics Letters A* **141**(8,9) 386-390.

Mach, J., Mas, F., and Saqués, F. [1995] 'Two representations in multifractal analysis,' *Journal of Physics A: Mathematics and General* **28** 5607-5622.

Mandelbrot, B. [1983] *The Fractal Geometry of Nature* (W. H. Freeman and Co., New York).

Meisel, L. V., M. Johnson, and P. J. Cote [1992] 'Box-counting multifractal analysis,' *Physical Review A* **45**(10), 6989-6996.

Meneveau, C., and K. R. Sreenivasan [1989] 'Measurement of $f(\alpha)$ from scaling histograms, and applications to dynamical systems and fully developed turbulence,' *Physics Letters A* **137**(3), 103-112.

Molteno, T. C. A. [1993] 'Fast $O(N)$ box-counting algorithm for estimating dimensions,' *Physical Review E* **48**(5), R3263-R3266.

Moon, F. C. [1992] *Chaotic and Fractal Dynamics* (Wiley Interscience, New York).

Muzy, J. F., Bacry, E. and Arneodo, A. [1994] 'The multifractal formalism revisited with wavelets,' *International Journal of Bifurcation and Chaos* **4**(2) 245-302.

Pastor-Satorras, R., and R. H. Riedi [1996] 'Numerical estimates of the generalized dimensions of the Hénon attractor for negative q,' *Journal of Physics A: Mathematical and General* **29**, L391-L398.

Pawelzik, K., and H. G. Schuster [1987] 'Generalized dimensions and entropies from a measured time series,' *Physical Review A* **35**(1), 481-484.

Rényi, A. [1970] *Probability Theory* (North-Holland Publishing Co., Amsterdam).

Takens, F. [1981] 'Detecting strange attractors in turbulence,' in D. A. Rand and L. S. Young (eds.), *Lecture Notes in Mathematics* **898** (Springer-Verlag, New York) 266-281.

Yamaguti, M., and Prado, C. [1995] 'A direct calculation of the spectrum of singularities $f(\alpha)$ of multifractals,' *Physics Letters A* **206** 318-322.

Yamaguti, M., and Prado, C. [1997] 'Smart covering for a box-counting algorithm,' *Physical Review E* **55**(6) 7726-7732.

Figure Captions

Figure 1. This schematic diagram shows the self similarity of the algorithm. There are four boxes in the first level, together covering the entire data set. Suppose there are data in the box labeled (0,0). Then, the algorithm zooms into this box, and divides it into the four boxes in the second level. In the case pictured, there are data in box (0,1), so we zoom in and divide it into four boxes at the third level. At this level, we see the resolution of the image.

Figure 2. A plot of cpu time vs. n (in thousands) shows that the algorithm is $O(n)$. These computations went through ten levels of zooming, with an output resolution of one box count per binary decade in r. Plus marks joined by a solid line are for tests of a one-dimensional line in a two-dimensional embedding space, and circles joined by a dashed line represent tests on uniformly filled two-dimensional space.

Figure 3. Dimensional calculations for the binomial multiplicative process with a probability of 0.25 associated with the left, and 0.75 with the right. (a) Rényi dimensions $D(q)$. (b) Dimension spectrum $f(\alpha)$. The calculations from box counting (circles) are compared to the theoretical (solid) curve (Feder, 1988). Resolution in r was one output per binary decade.

Figure 4. Scattering in the partition function $Z(q,r)$ for the case of $q = -5$ for low-resolution data (one r sample per binary decade).

Figure 5. High-resolution data ($\log r$ spaced at ten samples per binary decade) shows a scattering pattern.

Figure 6. Scattering in partition functions ($q = -5$) calculated for uniform one-dimensional sets indicates that the cause is in the partially covered edges of the set. These lead to partially filled boxes with low probabilities, not characteristic to the distribution in the set. Circles (o) represent the box-covering computations on 16K points, and plus signs ($+$) depict theoretical values for a solid line.

Figure 7. Partition functions ($q = -5$) for the binomial multiplicative process, calculated with a modified box count which omits uncharacteristic low-probability boxes.

Figure 8. For the modified box count which omits the uncharacteristic low-probability boxes, (a) the corresponding $D(q)$ curve (o), and (b) the corresponding $f(\alpha)$ curve (o), are compared to the corresponding theoretical (solid) curves.