

AD-A253 960



2

AD

TECHNICAL REPORT ARCCB-TR-92025

BOX-COUNTING MULTIFRACTAL ANALYSIS

L.V. MEISEL
M.A. JOHNSON
P.J. COTE

DTIC
ELECTE
AUG 13 1992
S A D

JUNE 1992



US ARMY ARMAMENT RESEARCH,
DEVELOPMENT AND ENGINEERING CENTER
CLOSE COMBAT ARMAMENTS CENTER
BENET LABORATORIES
WATERVLIET, N.Y. 12189-4050



APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

92-22748



92 8 11 090

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The use of trade name(s) and/or manufacturer(s) does not constitute an official indorsement or approval.

DESTRUCTION NOTICE

For classified documents, follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX.

For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.

For unclassified, unlimited documents, destroy when the report is no longer needed. Do not return it to the originator.

REPORT DOCUMENTATION PAGEForm Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 1992	3. REPORT TYPE AND DATES COVERED Final	
4. TITLE AND SUBTITLE BOX-COUNTING MULTIFRACTAL ANALYSIS			5. FUNDING NUMBERS AMCMS No. 6111.02.H610.0 PRON No. 1A14Z1CANMBJ	
6. AUTHOR(S) L.V. Meisel, M.A. Johnson, and P.J. Cote				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army ARDEC Benét Laboratories, SMCAR-CCB-TL Watervliet, NY 12189-4050			8. PERFORMING ORGANIZATION REPORT NUMBER ARCCB-TR-92025	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army ARDEC Close Combat Armaments Center Picatinny Arsenal, NJ 07806-5000			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES Presented at the ARO Workshop on Fractals, Army Research Office, Research Triangle Park, NC, 23 May 1991. Published in Physical Review A.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Two box-counting algorithms for the determination of generalized fractal dimensions are described. Results of application of the algorithm to Euclidean curves, quadric islands, Koch symmetric and asymmetric triadic snowflakes, and split snowflake halls are described. Comparison to analytic results for the model curves is provided, and the effectiveness of the algorithms is discussed.				
14. SUBJECT TERMS Multifractal, Fractal Analysis, Box-Counting			15. NUMBER OF PAGES 26	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

INTRODUCTION	1
THEORY	1
Fractal Measures	1
Box-Counting	2
Numerical Methods	2
The Sorting Box-Counting Algorithm	2
Discussion of the Sorting Algorithm	3
The Agglomeration Box-Counting Algorithm	5
Discussion of the Agglomeration Algorithm	5
RESULTS	5
Monofractals in E^2	5
Multifractals in E^1	7
Multifractals in E^2	7
CONCLUSIONS	9
REFERENCES	10

TABLES

Table 1. Normalized Agglomeration Box-Counting [$K=8, M=1$] Fractal Dimension $D_B(q)/D(q)$ versus Level for (Symmetric) Koch Triadic Snowflakes in E^2	11
Table 2. Normalized Agglomeration Box-Counting [$K=8, M=1$] Fractal Dimension $D_B(q)/D(q)$ versus Level for Asymmetric $[0.4, 0.2]$ Koch Triadic Snowflakes in E^2	11
Table 3. Normalized Agglomeration Box-Counting [$K=3, M=1, P=1$] Fractal Dimension $D_B(q)/D(q)$ versus Level for Asymmetric $[0.4, 0.2]$ Koch Triadic Snowflakes in E^2	12
Table 4. Normalized Agglomeration Box-Counting [$K=8, M=1$] Fractal Dimension $D_B(q)/D(q)$ versus Level for Split Snowflake Halls in E^2	13
Table 5. Normalized Agglomeration Box-Counting [$K=3, M=1, P=1$] Fractal Dimension $D_B(q)/D(q)$ versus level for split snowflake halls in E^2	13

LIST OF ILLUSTRATIONS

Figure 1. Agglomeration box-counting [$K=8, M=1$] fractal dimension $D_B(q)$ versus logarithm of number of points for random points on line segments in E^2 .	
(a) Segment oriented at 45° with respect to box edge	14
(b) Segment oriented at 18° with respect to box edge.	15
Figure 2. Agglomeration box-counting [$K=8, M=1$] fractal dimension $D_B(q)$ versus logarithm of number of points for random points on unit square in E^2	16

Figure 3. Sorting box-counting fractal dimension $D_B(q)$ versus q for various ϵ^* for 2000 random points on unit square in E^2	17
Figure 4. Agglomeration box-counting $[K=8, M=1]$ fractal dimension $D_B(q)$ versus logarithm of number of points for (symmetric) Koch triadic snowflakes in E^2	18
Figure 5. Sorting box-counting fractal dimension $D_B(q)$ versus q for various ϵ^* for asymmetric $[0.4, 0.2]$ Koch snowflakes in E^2	19
Figure 6. Agglomeration box-counting $[K=8, M=1]$ fractal dimension $D_B(q)$ versus logarithm of number of points for asymmetric $[0.4, 0.2]$ Koch snowflakes in E^2	20
Figure 7. Agglomeration box-counting $[K=4, M=1, P=1]$ fractal dimension $D_B(q)$ versus logarithm of number of points for asymmetric $[0.4, 0.2]$ Koch snowflakes in E^2	21
Figure 8. Sorting box-counting fractal dimension $D_B(q)$ versus q for various ϵ^* for level 4 ($N = 14,642$) split snowflake halls in E^2	22
Figure 9. Agglomeration box-counting $[K=8, M=1]$ fractal dimension $D_B(q)$ versus logarithm of number of points for split snowflake halls in E^2	23
Figure 10. Agglomeration box-counting $[K=3, M=1, P=1]$ fractal dimension $D_B(q)$ versus logarithm of number of points for split snowflake halls in E^2	24

DTIC QUALITY INSPECTED 5

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and for Special
A-1	

INTRODUCTION

The accomplishments of fractal modeling in materials to date are impressive. However, characterization of fractal structure in terms of the "fractal dimension" are incomplete. The "fractal dimension" tells us how the length, area, or density of an object varies with scale, but much more information is required to characterize or model the more complicated structures that occur in nature. The theory of multifractal objects provides a more comprehensive description of fractal objects. The major efforts in multifractal analysis have been directed toward the understanding of the subset of the more complicated fractal objects described in Halsey, Jensen, Kadanoff, Procaccia, and Shraiman (ref 1).

In order to optimally exploit multifractal perspectives, it is imperative that quantitative techniques for the measurements of fractal properties be developed. Box-counting techniques for the measurement of generalized fractal dimensions render the most direct application of the ideas in Halsey et al. (ref 1).

Block, Bloh, and Schellnhuber (ref 2) (BBS) have described a box-counting algorithm for the measurement of generalized (fractal) dimensions, which they refer to as "Efficient Box-Counting" (EBC). The present work describes two other realizations of box-counting procedures and addresses techniques for evaluating generalized fractal dimensions over wider ranges of the controlling parameters than that covered in the work of BBS. The multifractal models employed to test the effectiveness of the algorithms are subject to simple analysis along the lines in Halsey et al. (ref 1), so that our "measurements" are compared directly with analytic results.

THEORY

Fractal Measures

Although one can conceive of a more general class of fractal point sets than that analyzed in the seminal work of Halsey et al. (ref 1), it provides a basis for the analysis of most of the known fractal point sets, curves, etc. In particular, it provides the theoretical basis for multifractal analysis of the Koch constructions studied here. Following Halsey et al. (ref 1), define a partition function,

$$\Gamma(q, \gamma, L) = \sum_i \frac{p_i^q}{L_i^\gamma} \quad \text{where } \gamma = (1 - q) D_B(q). \quad (1)$$

Then, for recursive constructs,

$$\Gamma(q, \gamma, L^n) = [\Gamma(q, \gamma, L)]^n, \quad \text{so that } \Gamma(q, \gamma(q), L) = 1. \quad (2)$$

The generators of the multifractal models studied here have identical weights p_i , which sum to unity, and have two length scales. Letting n_i be the number of elements in the generator characterized by the length scale L_i , the intrinsic values of $D(q)$ are obtained by solving the transcendental equation

$$p^q [n_1 L_1^{-\gamma} + n_2 L_2^{-\gamma}] = 1 \quad (3)$$

for $\gamma(q) = (q-1) D_B(q)$. This is a trivial problem, however, one could avoid dealing with transcendental equations and solve

$$q(\gamma) = -\log [n_1 L_1^{-\gamma} + n_2 L_2^{-\gamma}] / \log [p] \quad (3a)$$

for $q(\gamma)$. For multifractals having many scales and weights, solution of a transcendental equation is required.

Box-Counting

Box-counting methods for the determination of generalized fractal dimensions are, in principle, well known. Recently, Gould and Tobochnik (ref 3) presented a valuable pedagogical review of the underlying concepts of box-counting methods such as those employed in the present study. The details of the numerical procedures employed here are provided below.

Numerical Methods

Two box-counting algorithms have been employed. The first is a sorting algorithm, which is similar to the EBC algorithm described in Reference 2. The second algorithm employs agglomeration of the results from a smallest set of boxes.

The following definitions are useful in the description of the box-counting algorithms:

- i. Let $S = \{s[j], j=1, \dots, N\}$ be a subset of points on the fractal set, where, in order to simplify the exposition, the origin is chosen in such a way that all the $s_\delta[i]$ are positive.

- ii. Let $\{u_\delta \mid \delta=1, d\}$ define an (arbitrary) orthogonal basis for the space containing S .

- iii. Define the largest scale:

$$E_{\max} = \max_{i,j,\delta} |u_\delta \cdot (s[i] - s[j])| \quad (4)$$

- iv. Map the fractal subset S into the unit hypercube (d -cube) Q , by means of the affine transformation (which does not change its fractal properties):

$$s^*[i] = \frac{s[i]}{E_{\max} + \epsilon} \quad (5)$$

where ϵ is a small positive number.

- v. Define the minimum normalized point set spacing

$$E_{\min} = \min_{i \neq j} |s^*[i] - s^*[j]|. \quad (6)$$

The Sorting Box-Counting Algorithm

- i. Define a set of hypercubes (boxes), having edges E_m , whose logarithms are uniformly spaced:

$$E_m = \frac{1}{\text{int}[R^{-1} (\epsilon^*)^{-\frac{m}{M}}]}, \quad m = 0, 1, \dots, M \quad (7)$$

The parameter R allows one to select normalized maximum box sizes less than unity. $R=1$ in EBC.

The parameter ϵ^* allows one to select minimum box edge values greater than the minimum normalized point set spacing, $E_{\min}$. The minimum box edge is (approximately) $E_{\min}$ in EBC.

- ii. Sort the $\{s^*[i]\}$ according to $s_1^*[i]$ using "Indexed HeapSort" (ref 4). Computation time for this step goes like $N \log(N)$.
- iii. For each E_m , compute the "partition function" $Z_m(q)$ as follows:

a. Compute a set of integer-valued coordinates:

$$\{a_\delta[i] = s_\delta^*[i] / E_m \mid \delta = 1, \dots, d, i = 1, \dots, N\} \quad (8)$$

N.b., The integer-valued coordinates will already be sorted according to $a_1[i]$.

b. Define sublists of the $a_\delta[i]$ corresponding to the different values of $a_1[i]$.

c. Compute the box occupation probabilities $p_n(E_m)$ by HeapSorting (ref 4) the sublists. When the topological dimension $d > 2$ sorting is accomplished by sequentially sorting or by sorting coordinates combined as in EBC, i.e., the $(d - 1) N$ integer coordinates are combined into N integers comprising a "linear list" according to the prescription:

$$\{f[i] \mid f[i] = \sum_{\delta=2}^d a_\delta[i] E_m^{2-\delta}, i = 1, \dots, N\} \quad (9)$$

N.b., The length of the sublists will be substantially smaller than N unless E_m is near unity, and when E_m is near unity, the number of boxes is small and the sorting is much faster than $N \log(N)$.

d. Compute the "partition function" $Z_m(q)$:

$$Z_m(q) = \sum_{n=1}^{N(E_m)} [p_n(E_m)]^q \quad (10)$$

where $N(E_m)$ is the number of E_m -boxes containing an element of S .

- iv. Compute the generalized box-counting fractal dimension $D_B(q)$, which is defined as the slope of the straight line fit to $\log[Z_m(q)]/(q-1)$ vs $\log(E_m)$. As in BBS, the subscript B makes explicit the distinction between the intrinsic generalized dimension $D(q)$ and that measured by application of box-counting to finite subsets of the given fractal set. The basis for this analysis is the generalized fractal dimension defined by Hentschel and Procaccia (ref 5):

$$D(q) = \frac{1}{q-1} \lim_{E_m \rightarrow 0} \frac{\log(Z_m(q))}{\log(E_m)} \quad (11)$$

where the limit is to be taken as E_m goes to zero for the perfect fractal. This form is also discussed in Halsey et al. (ref 1).

Discussion of the Sorting Algorithm

Sorting algorithms are flexible in the range and distribution of box sizes employed and are efficient in the use of storage. However, computation time goes like $N \log(N)$, and precise coordinate definition is required for proper box allocation at each L . We discuss the present sorting algorithm by comparison with EBC.

Most of the features of EBC pertain to efficient use of computer storage and computation time and do not affect the reliability of the determined generalized fractal dimensions. In particular, although different sorting strategies may result in different computation times, they cannot affect the results.

There are five features of the EBC procedure that are not in accord with the present procedure and/or may not be routine assumptions in box-counting and may influence the results.

- i. Although they do not disclose the essential features of the technique, BBS emphasize the fact that an essential feature of the EBC method is "classwise linear regression with random point selection" for the determination of the slope. "Classwise linear regression with random point selection" determines $D_B(q)$ for the subset ("pivotal values") of E_m (from the original logarithmically spaced set) for which "best" linear fitting is achieved.

The specific algorithm employed in EBC might make a substantive difference for $q < 0$, where plots of $\log[Z_m(q)]/(q-1)$ against $\log(E_m)$ tend to be extensively scattered. (BBS did not report $D_B(q)$ for $q < 0$.)

One expects that conventional linear regression is applied in most box-counting analyses. We have applied conventional linear regression based on minimization of least squares and absolute deviations in the present study.

- ii. EBC chooses the minimum box side in $\{E_m\}$ as the minimum magnitude of the vectorial distance between elements of the point set S . This prescription has the advantage of defining ϵ^* simply and uniquely in terms of the points in S . This assumption yielded unsatisfactory results when applied to the analysis of the multifractal point sets studied here.
- iii. EBC chooses the largest box size such that (within a small positive number ϵ) it just covers the point set in the arbitrarily selected orientation of the boxes. We have run calculations employing smaller "largest" E_m (i.e., we have run tests with $R < 1$). Although there are observable changes in $D_B(q)$ with R , the effects were small for $R \in [0.1, 1]$ in the present work.
- iv. Uniform spacing of $\log(E_m)$ is prescribed in EBC. This amounts to a choice of weighting factors for different levels of scaling. We have investigated other sets of $\{E_m\}$, which seem to work just as well. However, logarithmically spaced $\{E_m\}$ is a natural choice in the sense that each order of scales is equally represented, and so it is adopted for the present work.
- v. The use of QuickSort (ref 4) in EBC cannot lead to erroneous results and is usually the fastest sorting algorithm; however, it is dangerous in the sense that it can require computation times that go like N^2 (see Reference 5). Thus, we employ HeapSort (ref 4), for which computation time always goes like $N \log(N)$.

Although sorting of the derived "linear list" as in EBC is conceptually simple, it seems to be a bad strategy (in terms of computation time) for large lists and ϵ^* near E_{min} . For example, applying the present algorithm to a point set in E^1 , after the presort of the original (possibly floating point) list of coordinates, only searching of ordered lists of integers for changes is required, i.e., no further sorting is required in E^1 . Similar efficiencies associated with sorting shorter sublists can be expected for $d > 1$. At some value of ϵ^* , sufficiently larger than E_{min} , and/or for a small enough number of values of E_m , it becomes more efficient to avoid presorting of the entire list even in E^1 , and one expects that no penalty in computation time should be associated with the use of the derived "linear list."

The Agglomeration Box-Counting Algorithm

- i. Define a set of nested hypercubes ("boxes") appropriate for the given (or anticipated) point set S, such that the boxes have edge lengths,

$$E(k, m, \dots) = E_0 / (2^k 3^m \dots) \quad (12)$$

where $k \in \{0, 1, \dots, K\}$, $m \in \{0, 1, \dots, M\}$, ...

We refer to the smallest boxes in the set, i.e., those having edge length $E(K, M, \dots)$, as elementary hypercubes or elementary boxes.

- ii. Determine the occupation numbers $\langle n_i \rangle$ for each of the elementary hypercubes. If the boxes do not include all members of S, redefine N

$$N = \sum_i^{\text{elementary boxes}} \langle n_i \rangle.$$

- iii. For all combinations of $k, m, \dots$:

a. Compute the occupation numbers n_i for each of the $[2^k 3^m \dots]^d$ hypercubes by summing the occupation numbers of the constituent subsets of contained elementary hypercubes. We refer to this step as "agglomeration."

b. Determine the $p_i = n_i/N$ for each box and compute the "partition function" $Z(k, m, \dots; q)$ for the set of q for which $D_B(q)$ is to be determined as in Eq. (10). Empty boxes are not included.

- iv. Compute the generalized box-counting fractal dimension $D_B(q)$, which is defined as the slope of the straight line fit to $\log[Z(k, m, \dots; q)]/(q-1)$ vs $\log[E(k, m, \dots)]$.

Discussion of the Agglomeration Algorithm

The agglomeration algorithm may be inefficient in its use of storage, since information about empty boxes is maintained. However, the method has definite advantages. It is well-suited for application to "large" point sets, since computation time is essentially independent of N. Furthermore, the initial allocation of occupation values to the smallest boxes may be the natural way to treat the data obtained from automated image acquisition systems.

RESULTS

The present box-counting algorithms were applied to Euclidean point sets and point sets generated by simple Koch recursions (ref 6) in E^1 and E^2 . Applications of the sorting strategy in E^2 were for sets having $1,000 < N < 50,000$. The agglomeration technique was applied for $100 < N < 10^{11}$; sorting is not a practical alternative for the larger point sets.

Monofractals in E^2

A. Integer Dimensions; Euclidean Point Sets in E^2

Euclidean point sets are thought of as degenerate examples of monofractals in the present investigation.

1. Random Points on Lines and on the Unit Square

Figure 1 shows fractal dimension values obtained by applying the agglomeration algorithm for $K=8$ and $M=1$ to a range of numbers of random points on a straight line segment at two orientations in E^2 with respect to the box axes. $D_B(q)$ converges at all q ; but at negative q , $D_B(q)$ does not always converge to 1.0.

Figure 2 shows fractal dimension values obtained by applying the agglomeration algorithm for $K=8$ and $M=1$ to a range of numbers of random points on the unit square. Convergence of $D_B(q)$ within 1 percent to 2.0 is obtained at $q = -5, 0, 5$, and 25 ; $D_B(-25)$ converges to 2.062.

Figure 3 shows sorting results for 2000 random points in the plane for $\epsilon^* = \{0.01, 0.05, 0.10, 0.25\}$. The "best" values at negative q are obtained for $\epsilon^* \approx 0.05$ and at positive q for $\epsilon^* \approx 0.25$. The best values at all q would be obtained for ϵ^* between 0.05 and 0.10, which corresponds to an average box occupation of about 10 points. Better convergence is obtained with larger N , but the existence of an optimal ϵ^* substantially greater than the minimum spacing in S is generally found.

Similar results are obtained for random and equally spaced points on sawtooth curves and circular arcs.

B. Monofractal Koch Constructions in E^2

1. (Symmetric) Koch Triadic Snowflakes (ref 6)

Figure 4 shows results obtained by applying the agglomeration technique for $K=8$, $M=1$ to Koch triadic snowflakes.

a. Results at $q > 0$:

Although convergence is slower than that reported by BBS for EBC (on Euclidean point sets), agglomeration $D_B(q)$ exhibits convergence from below with increasing N ; $D_B(0)$, $D_B(5)$, and $D_B(25)$ are within 1 percent of $D(q)$ for $\log(N) > 4$.

Sorting box-counting results for the triadic snowflake were similar to those obtained for uniformly spaced Euclidean curves in that, for $2 E_{\min} < \epsilon^* < 0.05$, $0 < q < 9$ and $N > 1,000$, the $D_B(q)$ are weakly ϵ^* -dependent and are within about 2 percent of the intrinsic (constant) $D(q)$ values; $D_B(0)$ and $D_B(3)$ values within 0.5 percent of $D(0)$ were obtained for $\epsilon^* \approx 2 E_{\min}$ and $N = 3 \cdot 4^7 = 49,152$. $D_B(q)$ tended to decrease with increasing q for $q > 5$.

b. Results at $q < 0$:

The values for $D_B(-25)$ and $D_B(-5)$ obtained by the agglomeration technique for $K=8$, $M=1$ do not converge for $N < 10^5$; the "best" values are obtained when $\log(N) \approx 5.5$. The minimal box edge, $E(8,1) = E_{\max}/(2^8 \cdot 3) = E_{\max}/768$ corresponds to the minimum spacing at

$$\log(N) = \log(3) + D \log(768) \approx 4.1.$$

Thus, for $q < 0$, the best values obtain for minimal box edge about 10 times larger than the minimal spacing in the point set. The $D_B(-5)$ and $D_B(-25)$ curves cross near $\log(N) = 4.1$. Table 1 gives values of normalized $D_B(q)$ ($= D_B(q) \log(3)/\log(4)$) vs level ($N = 3 \cdot 4^{\text{level}}$) and q for the Koch triadic snowflakes.

For $q < 0$, $D_B(q)$ curves obtained by sorting were strongly dependent on ϵ^* and tended to more closely resemble the intrinsic (constant) $D(q)$ for $\epsilon^* \approx 2 E_{\min}$. For $\epsilon^* \leq E_{\min}$, $D_B(q)$ are monotonic increasing at negative q .

Similar results are obtained for Koch quadric islands (ref 6).

Multifractals in E^1

The sorting box-counting algorithm was applied to the logistic equation mapping employing one-dimensional hypercubes. The $f-\alpha$ spectra deduced from $D_B(q)$ were in accord with those shown in Gould and Tobochnik (ref 3).

Multifractals in E^2

The box-counting algorithms were applied to asymmetric Koch triadic snowflakes and Koch split snowflake halls. The asymmetric Koch triadic snowflakes were produced by modifying the symmetric Koch triadic snowflake generator such that the angles are unchanged, but the outer segments are twice as large as the central segments, i.e., the inner and outer segments of the generator have relative lengths 0.2 and 0.4, respectively. Mandelbrot (ref 6) defines split snowflake halls as the level 4 form of the construction, which at level 3, he defines as the monkey tree. Here we refer to all levels of the construction as split snowflake halls.

A. Asymmetric Koch Snowflakes

1. Sorting Results

Figure 5 shows $D_B(q)$ obtained by sorting box-counting for level 7 ($N = 49,152$), asymmetric $[0.4, 0.2]$ Koch snowflakes. The dashed curve is the intrinsic $D(q)$. The following attributes may be observed:

a. Results for $5 > q > 0$:

For ϵ^* values ranging between about 0.001 ($\approx 75 E_{\min}$) and about 0.010 ($\approx 750 E_{\min}$), $D_B(q)$ closely approximates the intrinsic $D(q)$. $D_B(0)$ values are weakly ϵ^* -dependent over the specified range of ϵ^* and are within 1 percent of their intrinsic values. $D_B(3)$ values are weakly ϵ^* -dependent over the specified range of ϵ^* and are generally within 2 percent of their intrinsic values. The discrepancy is larger at larger q . The discrepancy between $D_B(q)$ and $D(q)$ is quite large for $\epsilon^* = E_{\min}$.

The requirement that $\epsilon^* > 75 E_{\min}$ can be understood as follows: $E_{\min}$ is determined by the most dense regions of the point set. The snowflake considered has least dense regions, which are $(0.4/0.2)^2 = 128$ times less dense. Therefore, $\epsilon^* = 128 E_{\min}$ are minimal spacings in the least dense regions of the level 7 construct. Hence, $\epsilon^* = 75 E_{\min}$ is a reasonable value to characterize the entire point set, and it is clear that the EBC choice, $\epsilon^* = E_{\min}$, is too small.

b. Results at $q < 0$:

$D_B(q)$ curves were strongly dependent on ϵ^* . Curves that approximate $D(q)$ reasonably well over the entire range of q are obtained for ϵ^* between 0.001 and 0.004. The curve obtained for $\epsilon^* = 0.0025$ is closest to $D(q)$, while that for 0.001 is smoother near $q=0$ but is slightly further from $D(q)$. One may observe the beginning of the formation of a slight belly near $q = 0$, as ϵ^* increases through 0.0025.

2. Agglomeration Results

Figure 6 shows $D_B(q)$ curves for $q = \{-25, -5, 0, 5, 25\}$ and various N for the $[0.4, 0.2]$ asymmetric triadic snowflakes measured via $K = 8$, $M = 1$ boxes; Table 2 gives values of normalized $D_B(q)$ ($= D_B(q)/D(q)$) vs level ($N = 3 \cdot 4^{\text{level}}$) and q . Converged values for $q \geq 0$ are obtained as 0.4^{level} passes through the box edge length $E(8,1)$ of the elemental boxes. The "best" values at negative q occur for these transitional cases, which fall at $\log(N)$ slightly beyond that for which the $D_B(-5)$ and $D_B(-25)$ curves cross.

Figure 7 shows similar results for $K = 3$, $M = 1$, $P = 1$ ($E(3,1,1) = E_{\text{max}}/(2^3 \cdot 3 \cdot 5) = E_{\text{max}}/120$); Table 3 gives values of normalized $D_B(q)$ vs level and q . Note, in particular, that convergence is obtained for values of N about one-fifth as large as in the $K = 8$, $M = 1$ case.

B. Split Snowflake Halls

1. Sorting Results

Figure 8 shows $D_B(q)$ obtained by sorting box-counting for level 4 ($N = 1 + 11^4$) split snowflake halls. The dashed curve is the intrinsic $D(q)$. The following features may be observed:

a. Results for $q > 3$:

For ϵ' values ranging between about $7 E_{\text{min}}$ and $30 E_{\text{min}}$, $D_B(q)$ closely approximates the intrinsic $D(q)$. $D_B(3)$ values are weakly ϵ' -dependent over the specified range of ϵ' , but are generally within 2 percent of their intrinsic values and are closer at larger q . The discrepancy between $D_B(q)$ and $D(q)$ is large for $\epsilon' = E_{\text{min}}$.

b. Results for $3 > q > -1$:

For ϵ' values ranging between about $7 E_{\text{min}}$ and 0.07 , $D_B(q)$ are weakly ϵ' -dependent, but lie substantially below the intrinsic $D(q)$. $D_B(0)$ values are about 12 percent low.

c. Results at $q < -1$:

$D_B(q)$ curves were strongly dependent on ϵ' . Curves that most closely approximated $D(q)$ for this range of q , in the sense of rms deviation, behaved badly near $q = 0$ and had ϵ' values outside the range and larger than those for which the "best" approximation in the $q > -1$ region obtains. No value of ϵ' produced a monotonic decreasing $D_B(q)$.

2. Agglomeration Results

Figure 9 shows $D_B(q)$ curves for $q = \{-25, -5, 0, 5, 25\}$ and various N for split snowflake halls measured via $K = 8$, $M = 1$ boxes; Table 4 gives values of normalized $D_B(q)$ ($= D_B(q)/D(q)$) vs level ($N = 1 + 11^{\text{level}}$) and q . The "best" values at negative q occur for $\log(N)$ slightly larger than that for which the $D_B(-5)$ and $D_B(-25)$ curves cross.

Figure 10 shows similar results for $K = 3$, $M = 1$, $P = 1$; Table 5 gives values of normalized $D_B(q)$ vs level and q . Note, in particular, that convergence is obtained for values of N about one-fifth as large as in the $K = 8$, $M = 1$ case.

CONCLUSIONS

The sorting and agglomeration box-counting algorithms are effective for the determination of $D_B(q)$ for $q \geq 0$. However, the present implementations yield unreliable results for $q < 0$.

$D_B(q)$ values converged from below for $q \geq 0$ and overshoot and converged from above at $q < 0$ for random points on Euclidean objects. Therefore, convergence to $D(q)$ for Euclidean point sets is not a sufficient condition for demonstrating the effectiveness of an algorithm.

Sorting box-counting algorithms require storage of $d \cdot N$ double precision (eight bytes) floating point numbers and cpu time varies as $N \log(N)$. The input data to the algorithm are the coordinates of the points. In order to achieve the best results, the particle coordinates must be known precisely so that box occupancies for the various box edge lengths in the set can be determined. This constraint is relatively easy to satisfy for mappings or other recursively defined point sets, but may be exacting for experimentally acquired point set coordinates.

The agglomeration box-counting algorithm requires storage of $(2^K 3^M \dots)^d$ integers and the cpu time is proportional to $(2^K 3^M \dots)^d$. Although we used four byte integers in the present work, two byte integers are probably sufficient in practice. Cpu time and storage requirements are independent of N , and therefore agglomeration algorithms may be the only practical technique for application to cases where N is "large." The input data to the algorithm are elementary box occupation numbers rather than precise coordinates. This could be an important distinction for image acquisition systems, which provide data in this form. Thus, the agglomeration technique works equally well for mappings or other recursively defined point sets and for experimentally acquired point sets.

The values of N required for convergence of D at $q \geq 0$ were approximately the same for the sorting and agglomeration box-counting algorithms in the present investigation.

For example, for $d=2$ and $D(0) < 1.5$, our experience indicates that $N \approx 10^4$ is required for convergence. In this case, storage requirements for $(K=3, M=1, P=1)$ agglomeration (four byte integers) is approximately 60 Kbytes, while sorting requires 200 Kbytes. Larger N and number of elementary boxes are required for larger $D(0)$, but the relative advantage of the agglomeration technique with regard to storage is maintained.

We have not tested convergence of box-counting techniques for $d > 2$, however we anticipate that substantially larger N will be required for convergence. One might speculate that for $d=3$, $N \approx (10^4)^{1.5} = 10^6$, agglomeration $(K=3, L=1, M=1)$ uses 7 Mbytes. Sorting would require 25 Mbytes. Of course, if convergence is obtained for $N \approx 10^4$ for $d=3$, then sorting would require only 0.25 Mbytes, while agglomeration would still require 7 Mbytes.

Sorting box-counting algorithms might be the best choice for relatively small sets of precisely defined points. If "classwise linear regression with random point selection" implicitly yields sets of E_m , which give good values for $D_B(q)$ at $q < 0$, then EBC is probably best for the cases to which it may be applied. However, if this is the case, the principle can probably be applied to refine the agglomeration technique as well.

Agglomeration box-counting resulted in values for $D_B(q)$ at least as good as those determined by sorting for all cases studied here. For the large point sets employed in our convergence studies, sorting would have been (for the present) impractical. Computation times for the moderate-sized point sets (required for convergence) would be competitive; agglomeration, of course, becomes relatively more efficient as N increases.

REFERENCES

1. Thomas C. Halsey, Mogens H. Jensen, Leo P. Kadanoff, Itamar Procaccia, and Boris I. Shraiman *Physical Review A*, Vol. 33, 1986, p. 1141.
2. A. Block, W. von Bloh, and H.J. Schellnhuber, *Physical Review A*, Vol. 42, 1990, p. 1869 (BBS). Other recent references to sorting algorithms for the determination of $D(0)$ include: L.S. Liebovitch and T. Toth, *Physical Letters A*, Vol. 141, 1989, p. 386; X. J. Hou, R. G. Gilmore, G. B. Mindlin, and H. G. Solari, *Physical Letters A*, Vol. 151, 1990, p. 43.
3. Harvey Gould and Jan Tobochnik, *Computers in Physics* Vol. 4, 1990, p. 202.
4. William H. Press, Brian P. Flannery, Saul A. Teukolsky and William T. Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge, 1986.
5. H. G. E. Hentschel and I. Procaccia, *Physica D*, Vol. 8, 1983, p. 435.
6. B. B. Mandelbrot, *Fractal Geometry of Nature*, Freeman, New York, 1983.
7. For example, B. B. Mandelbrot, C. J. G. Evertsz, and Y. Hayakawa, *Physical Review A*, Vol. 42, 1990, p. 4528.

Table 1. Normalized Agglomeration Box-Counting [K=8,M=1] Fractal Dimension $D_B(q)/D(q)$ versus Level for (Symmetric) Koch Triadic Snowflakes in E^2 .
($N = 3 \cdot 4^{\text{LEVEL}}$)

Level	q=-25	q=-5	q=0	q=5	q=25
2	0.1187	0.1601	0.2435	0.2894	0.3191
4	0.4072	0.4993	0.7523	0.8004	0.8028
6	0.6511	0.7107	1.0070	0.9955	0.9688
8	1.0017	0.9581	1.0158	0.9985	0.9835
10	1.3632	1.2259	1.0161	0.9984	0.9834
12	1.7321	1.5283	1.0162	0.9983	0.9834
14	2.1031	1.8391	1.0162	0.9983	0.9834

Table 2. Normalized Agglomeration Box-Counting [K=8,M=1] Fractal Dimension $D_B(q)/D(q)$ versus Level for Asymmetric [0.4,0.2] Koch Triadic Snowflakes in E^2 .
($N = 3 \cdot 4^{\text{LEVEL}}$)

Level	q=-25	q=-5	q=0	q=5	q=25
2	0.1029	0.1548	0.2986	0.4298	0.5075
4	0.3862	0.4917	0.8062	0.9193	0.8707
6	0.7162	0.7859	1.0053	0.9577	0.9113
8	1.0580	1.0588	1.0210	0.9581	0.9115
10	1.4027	1.3428	1.0220	0.9580	0.9115
12	1.7496	1.6372	1.0221	0.9580	0.9115
14	2.0992	1.9445	1.0221	0.9580	0.9115

Table 3. Normalized Agglomeration Box-Counting [$K=3, M=1, P=1$] Fractal Dimension $D_B(q)/D(q)$ versus Level for Asymmetric $[0.4, 0.2]$ Koch Triadic Snowflakes in E^2 .
 $(N = 3 \cdot 4^{\text{LEVEL}})$

Level	$q=-25$	$q=-5$	$q=0$	$q=5$	$q=25$
1	0.0543	0.0824	0.1483	0.2630	0.3497
2	0.1834	0.2667	0.5045	0.6941	0.7652
3	0.3949	0.5077	0.8244	0.9155	0.8637
4	0.6178	0.7079	0.9773	0.9646	0.9083
5	0.8437	0.8819	1.0177	0.9737	0.9177
6	1.0930	1.0836	1.0246	0.9750	0.9198
7	1.3419	1.2949	1.0258	0.9753	0.9201
8	1.5448	1.4492	1.0261	0.9751	0.9199
9	1.7295	1.6108	1.0263	0.9751	0.9199
10	1.8953	1.7611	1.0263	0.9751	0.9199
11	2.0597	1.9090	1.0263	0.9751	0.9199
12	2.2287	2.0656	1.0263	0.9751	0.9199
13	2.3981	2.2227	1.0263	0.9751	0.9199
14	2.5674	2.3795	1.0263	0.9751	0.9199

Table 4. Normalized Agglomeration Box-Counting [K=8,M=1] Fractal Dimension $D_B(q)/D(q)$ versus Level for Split Snowflake Halls in E^2 .
($N = 1 + 11^{\text{LEVEL}}$)

Level	q=-25	q=-5	q=0	q=5	q=25
2	0.1203	0.1628	0.2747	0.4068	0.4678
3	0.2464	0.3226	0.5576	0.7343	0.7767
4	0.3764	0.4559	0.7908	0.9319	0.9135
5	0.6045	0.6526	0.9022	0.9748	0.9609
6	0.8599	0.8533	0.9303	0.9772	0.9688
7	1.0927	1.0349	0.9352	0.9771	0.9689
8	1.3575	1.2493	0.9366	0.9771	0.9689
9	1.6121	1.4718	0.9370	0.9771	0.9689
10	1.8223	1.6595	0.9372	0.9771	0.9688

Table 5. Normalized Agglomeration Box-Counting [K=3,M=1,P=1] Fractal Dimension $D_B(q)/D(q)$ versus level for split snowflake halls in E^2 .
($N = 1 + 11^{\text{LEVEL}}$)

Level	q=-25	q=-5	q=0	q=5	q=25
1	0.0091	0.0401	0.1128	0.1931	0.2338
2	0.2064	0.2742	0.4645	0.6393	0.6693
3	0.4098	0.4846	0.7682	0.9003	0.8818
4	0.6772	0.7033	0.8799	0.9531	0.9378
5	1.0265	0.9744	0.9016	0.9579	0.9490
6	1.2935	1.1909	0.9061	0.9571	0.9478
7	1.5803	1.4236	0.9070	0.9572	0.9479
8	1.7172	1.5409	0.9073	0.9573	0.9480
9	1.8404	1.6545	0.9075	0.9573	0.9480

For all Agglomeration Box-Counting figures:
 Open squares: $q = -25$; Triangles: $q = -5$; Circles: $q = 0$; Stars: $q = 5$; Filled Squares: $q = 25$

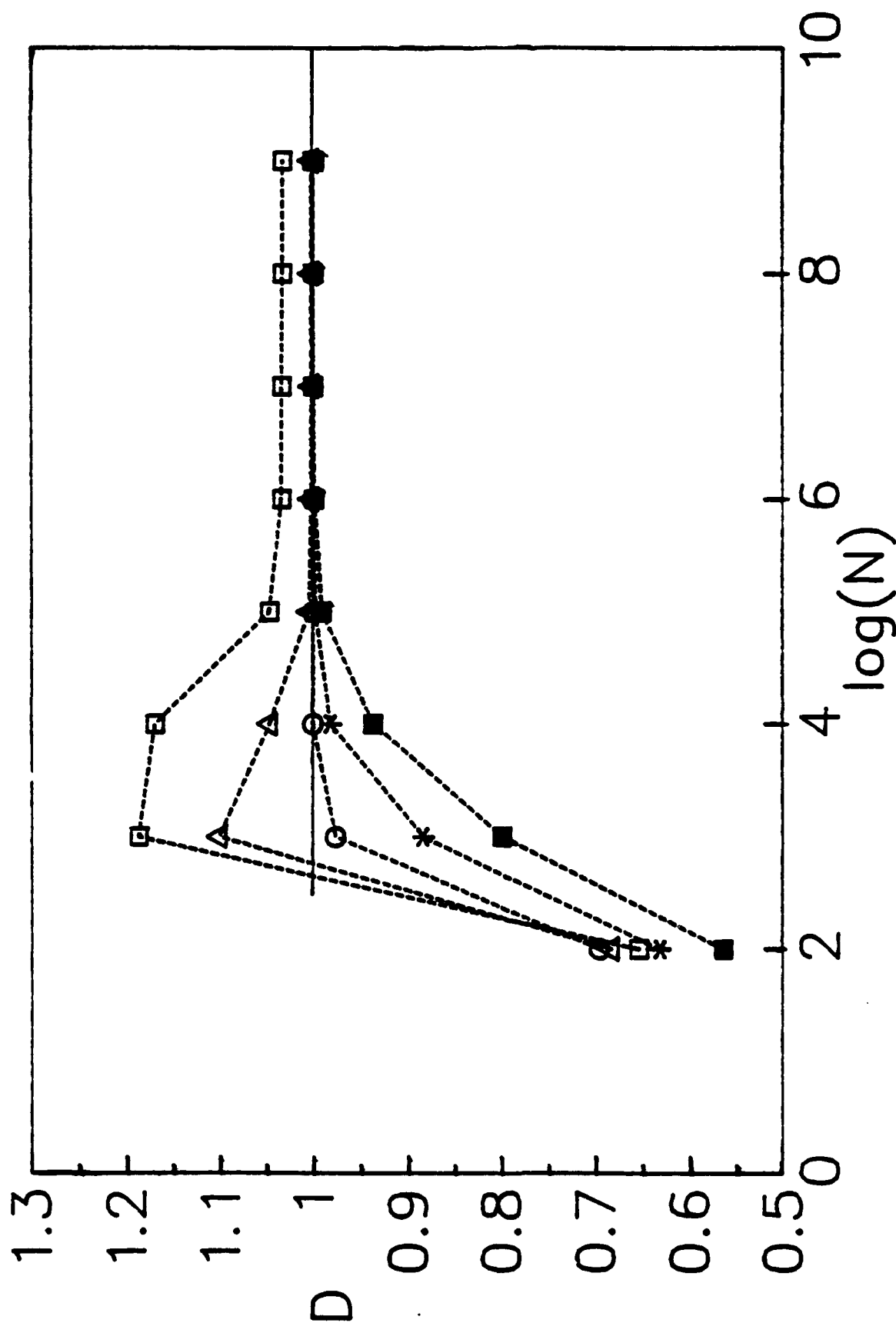


Figure 1. Agglomeration box-counting [$K=8, M=1$] fractal dimension $D_b(q)$ versus logarithm of number of points for random points on line segments in E^2 . (a) Segment oriented at 45° with respect to box edge.

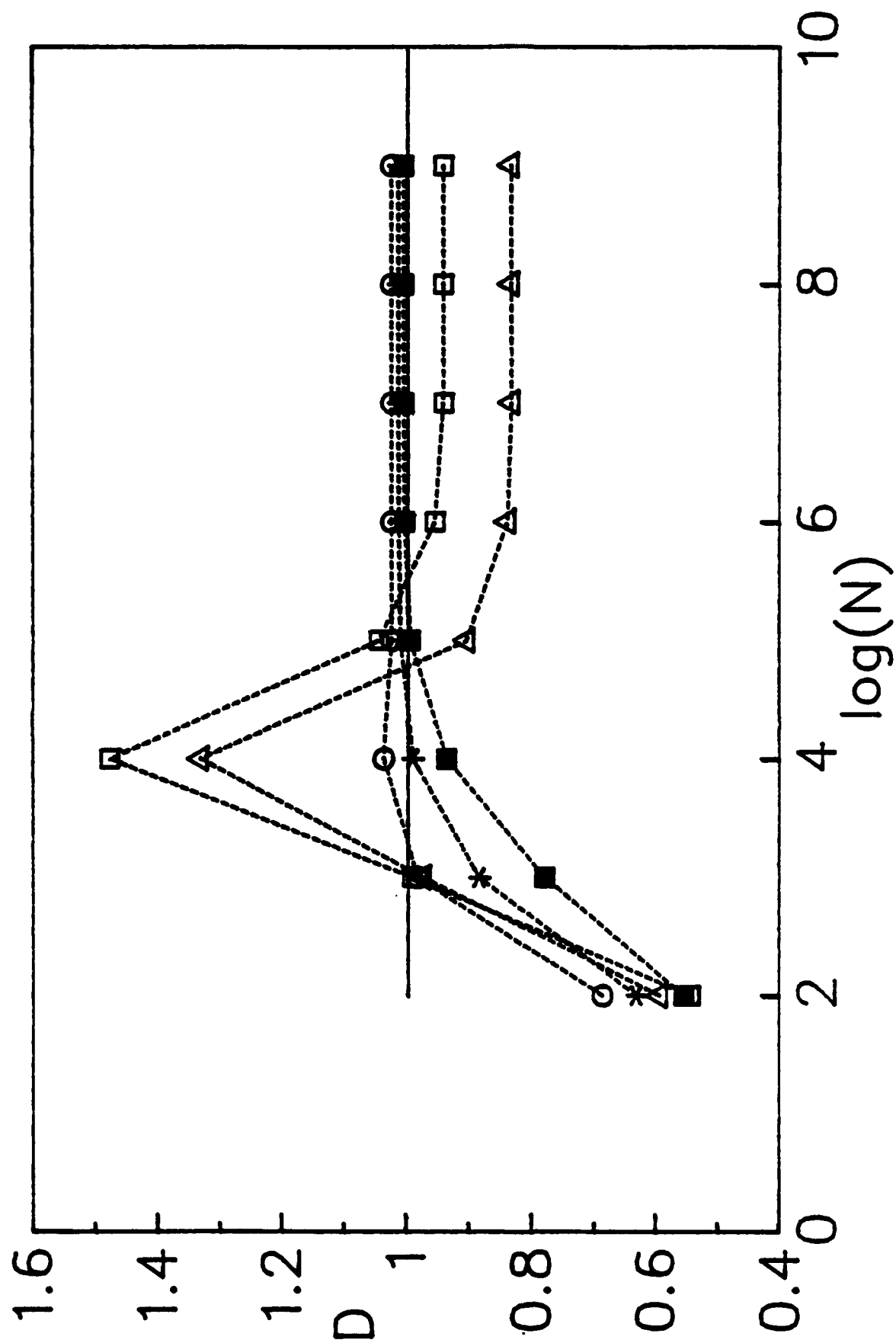


Figure 1 (b). Segment oriented at 18° with respect to box edge.

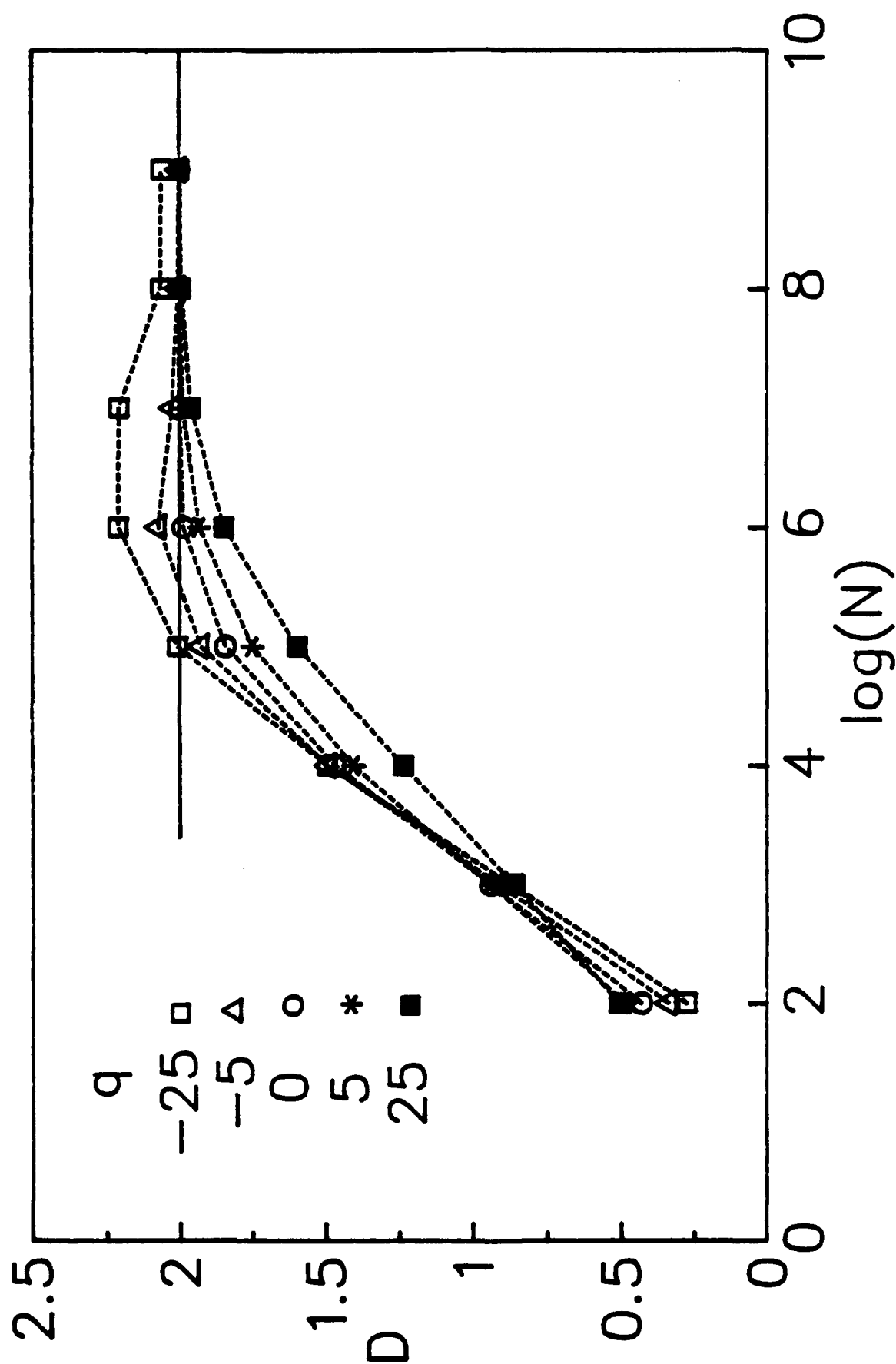


Figure 2. Agglomeration box-counting [$K=8, M=1$] fractal dimension $D_u(q)$ versus logarithm of number of points for random points on unit square in $\mathbb{R}^2$.

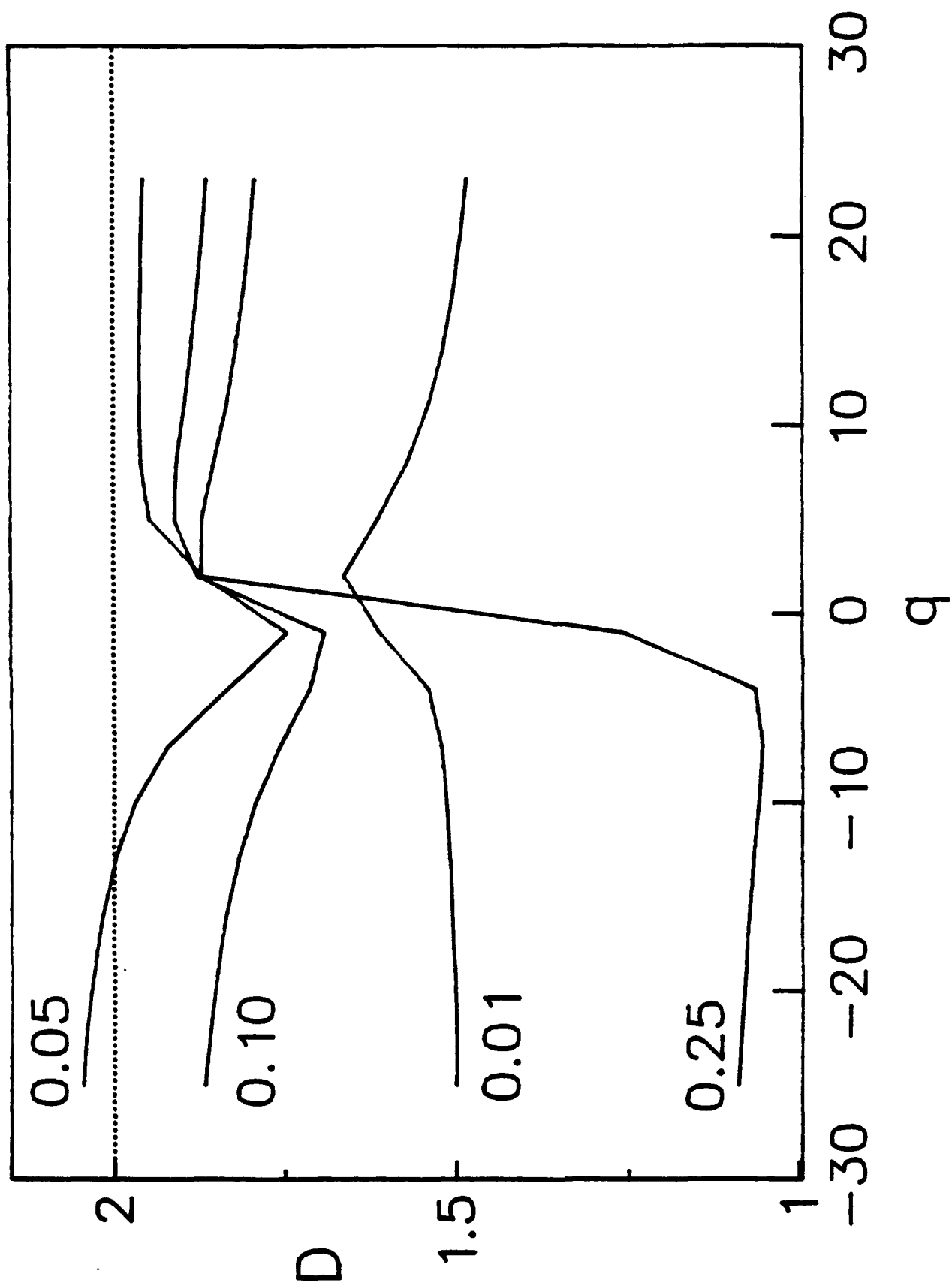


Figure 3. Sorting box-counting fractal dimension $D_B(q)$ versus q for various ϵ for 2000 random points on unit square in $\mathbb{R}^2$. Parameters are ϵ values for all sorting box-counting figures.

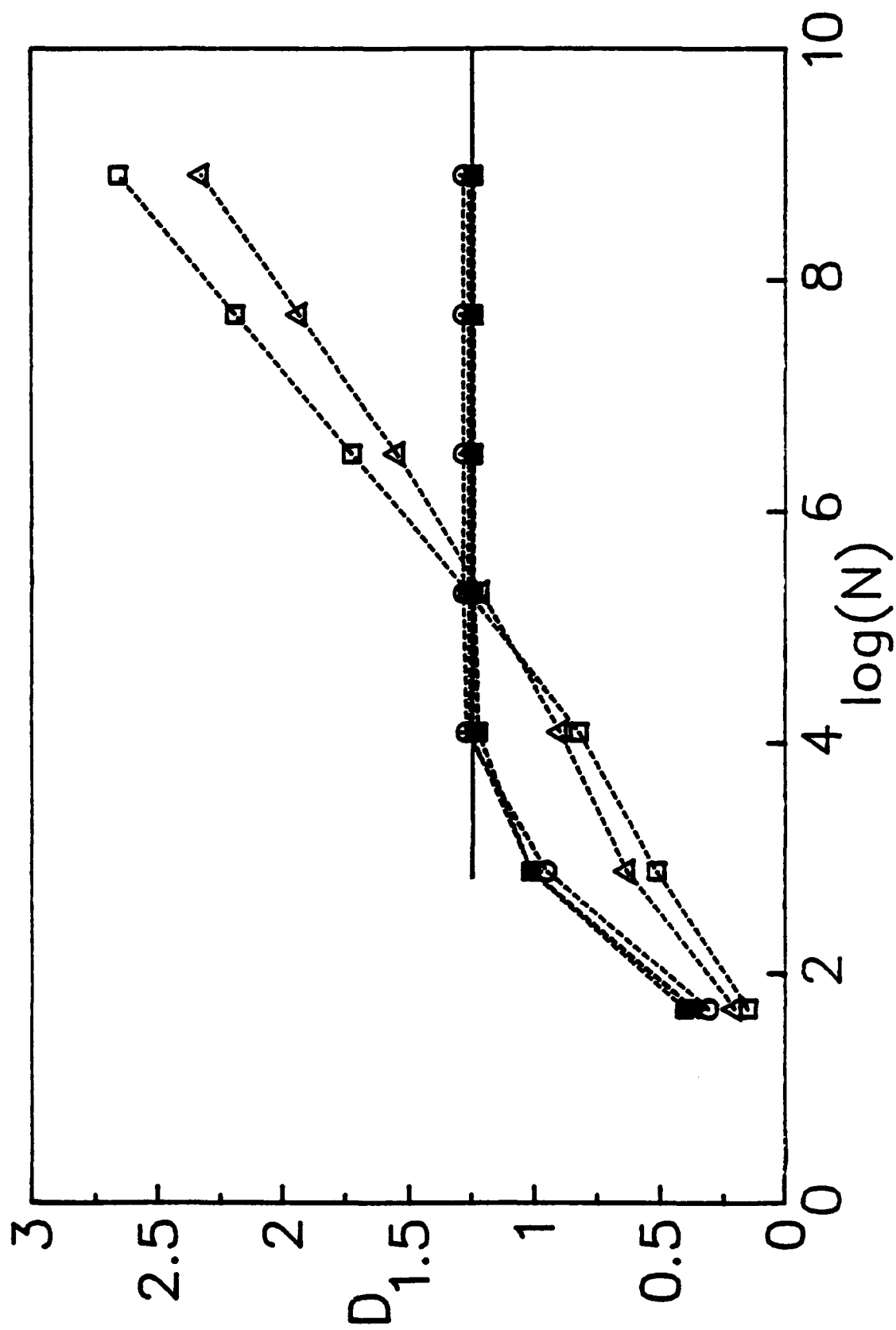


Figure 4. Agglomeration box-counting [$K=8, M=1$] fractal dimension $D_B(q)$ versus logarithm of number of points for (symmetric) Koch triadic snowflakes in $\mathbb{R}^2$.

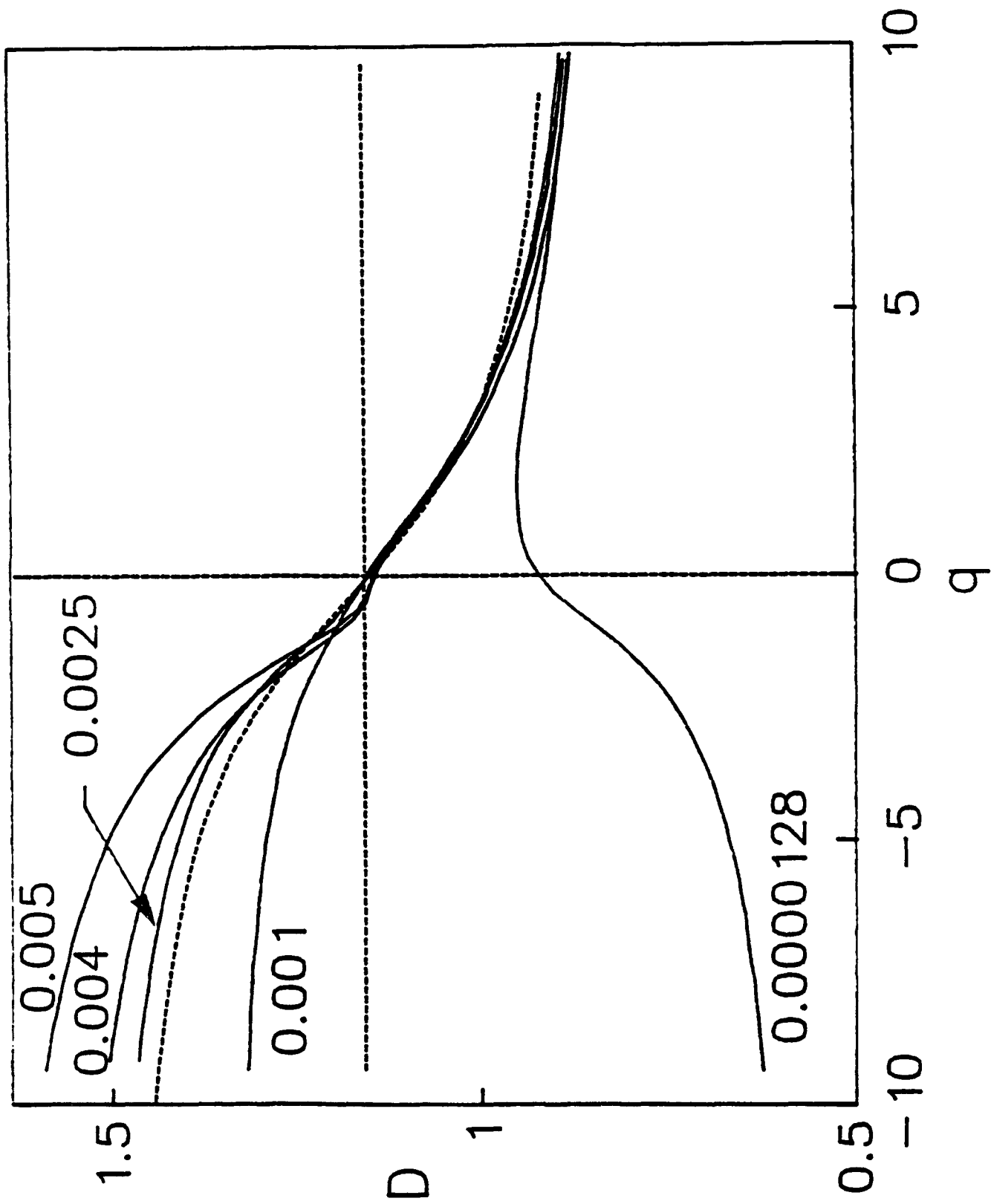


Figure 5. Sorting box-counting fractal dimension $D_a(q)$ versus q for various ϵ' for asymmetric $[0.4, 0.2]$ Koch snowflakes in E^2 . $N = 49, 152$. The dashed curve is the analytic result.

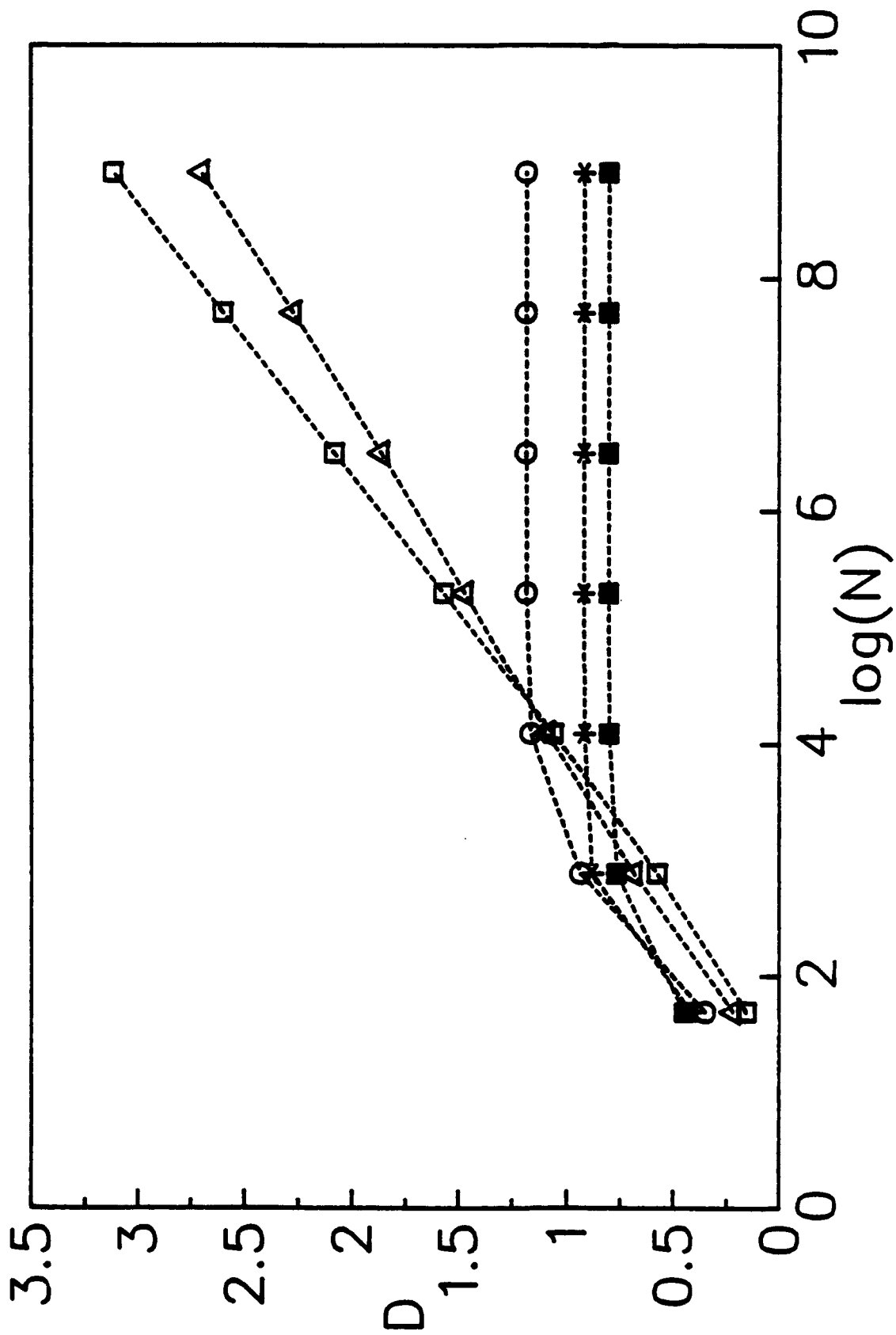


Fig. 6. Agglomeration box-counting $[K=8, M=1]$ fractal dimension $D_B(q)$ versus logarithm of number of points for asymmetric $[0.4, 0.2]$ Koch snowflakes in E^2 .

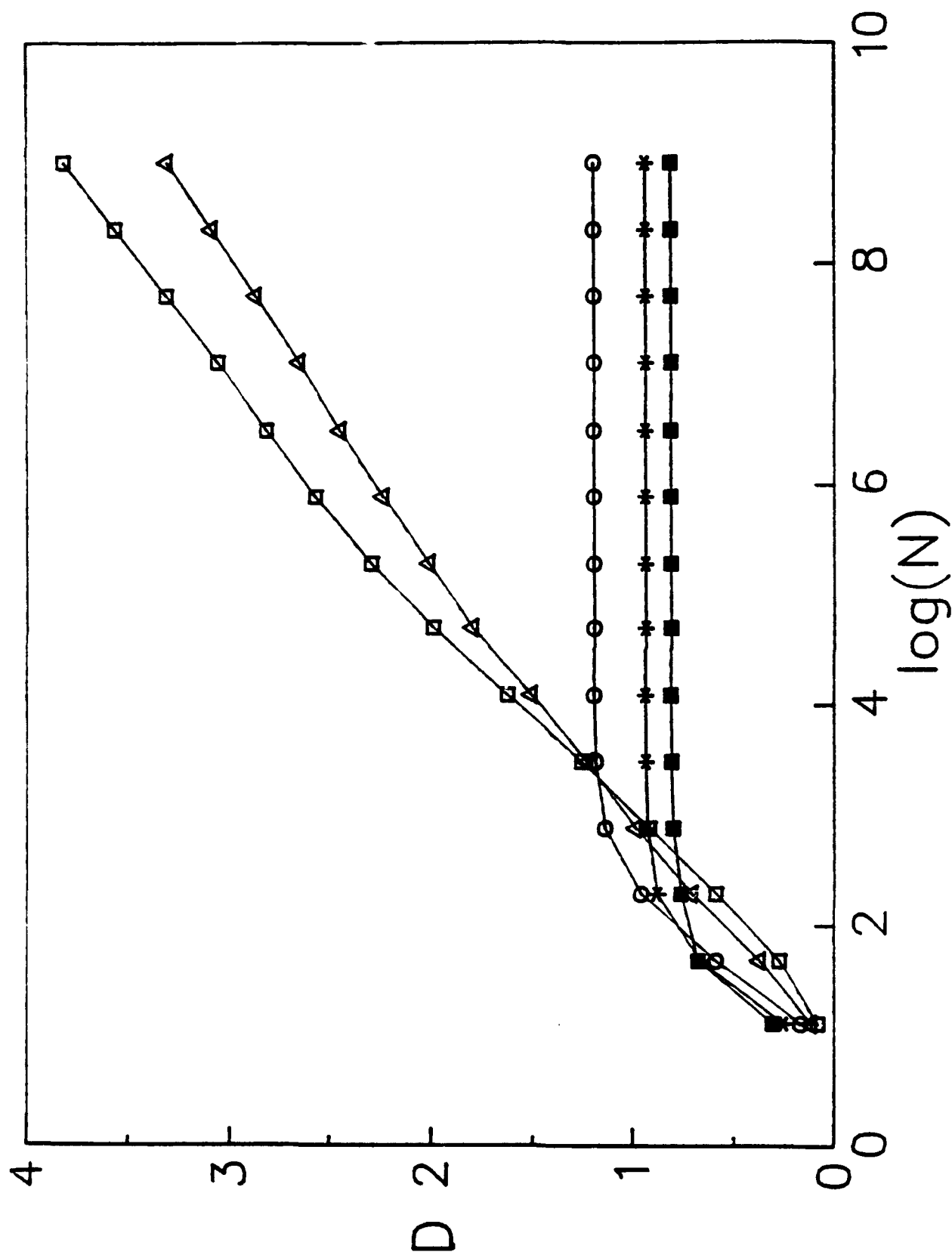


Figure 7. Agglomeration box-counting [$K=4, M=1, P=1$] fractal dimension $D_b(q)$ versus logarithm of number of points for asymmetric $[0.4, 0.2]$ Koch snowflakes in $\mathbb{R}^2$.

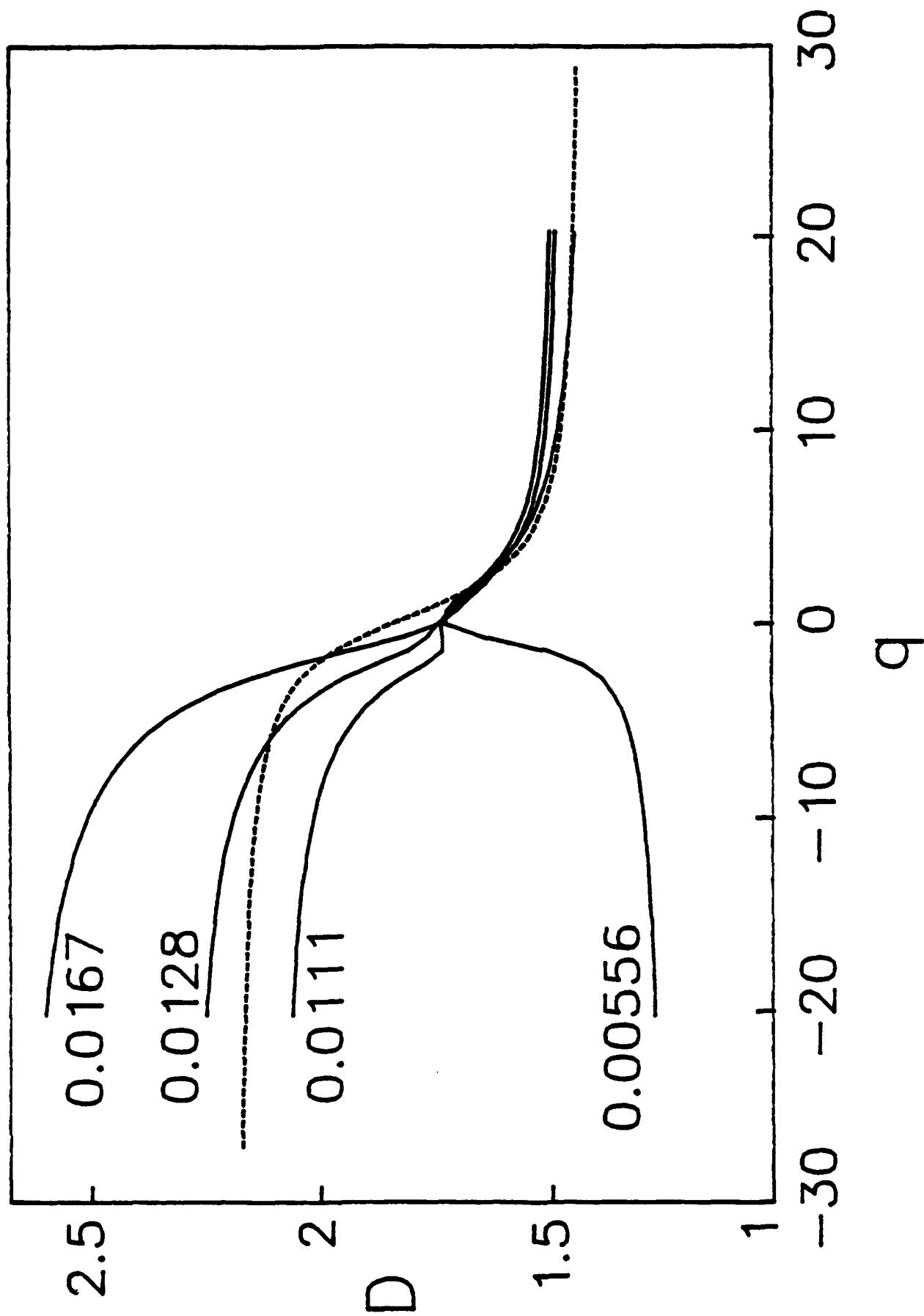


Figure 8. Sorting box-counting fractal dimension $D_a(q)$ versus q for various ϵ' for level 4 ($N = 14,642$) split snowflake halls in I^2 . The dashed curve is the analytic result.

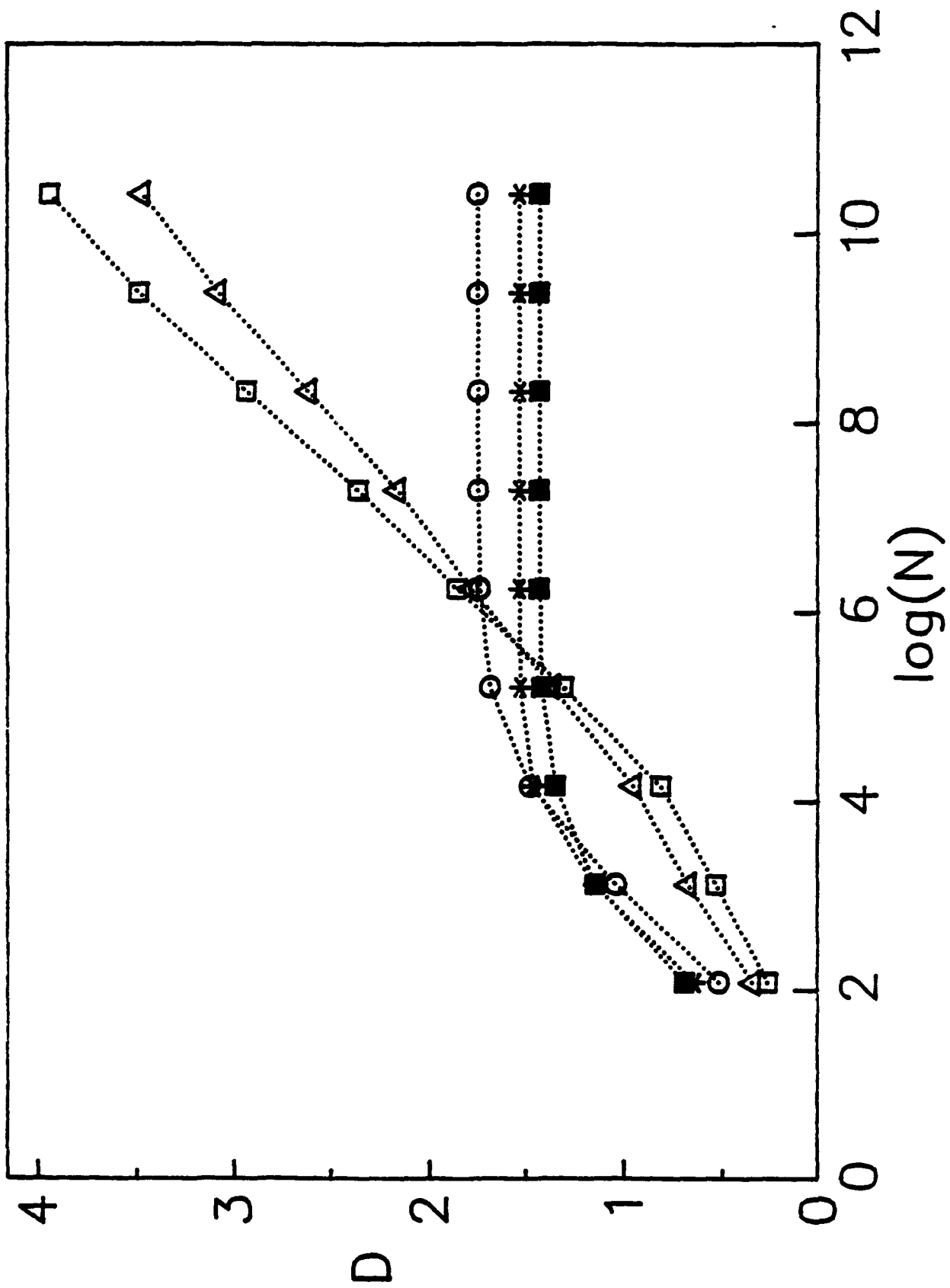


Figure 9. Agglomeration box-counting [$K=8, M=1$] fractal dimension $D_B(q)$ versus logarithm of number of points for split snowflake hulls in E^2 .

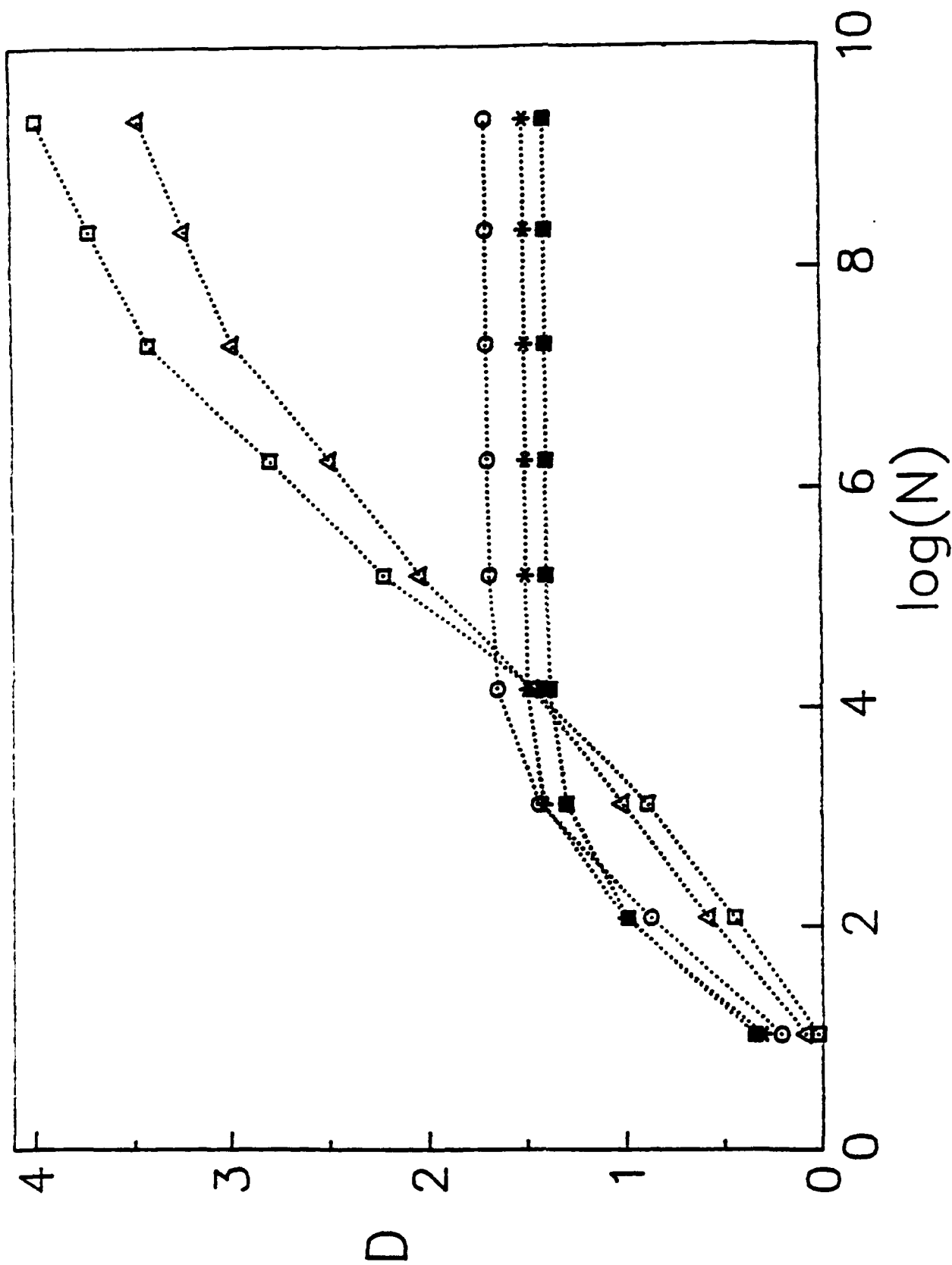


Figure 10. Agglomeration box-counting [$K=3, M=1, P=1$] fractal dimension $D_B(q)$ versus logarithm of number of points for split snowflake hulls in E^2 .

TECHNICAL REPORT INTERNAL DISTRIBUTION LIST

	NO. OF COPIES
CHIEF. DEVELOPMENT ENGINEERING DIVISION	
ATTN: SMCAR-CCB-DA	1
-DC	1
-DI	1
-DR	1
-DS (SYSTEMS)	1
CHIEF. ENGINEERING SUPPORT DIVISION	
ATTN: SMCAR-CCB-S	1
-SD	1
-SE	1
CHIEF. RESEARCH DIVISION	
ATTN: SMCAR-CCB-R	2
-RA	1
-RE	1
-RM	1
-RP	1
-RT	1
TECHNICAL LIBRARY	5
ATTN: SMCAR-CCB-TL	
TECHNICAL PUBLICATIONS & EDITING SECTION	3
ATTN: SMCAR-CCB-TL	
OPERATIONS DIRECTORATE	1
ATTN: SMCWV-ODP-P	
DIRECTOR. PROCUREMENT DIRECTORATE	1
ATTN: SMCWV-PP	
DIRECTOR. PRODUCT ASSURANCE DIRECTORATE	1
ATTN: SMCWV-OA	

NOTE: PLEASE NOTIFY DIRECTOR, BENET LABORATORIES, ATTN: SMCAR-CCB-TL, OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST

	<u>NO. OF COPIES</u>		<u>NO. OF COPIES</u>
ASST SEC OF THE ARMY RESEARCH AND DEVELOPMENT ATTN: DEPT FOR SCI AND TECH THE PENTAGON WASHINGTON, D.C. 20310-0103	1	COMMANDER ROCK ISLAND ARSENAL ATTN: SMCRI-ENM ROCK ISLAND, IL 61299-5000	1
ADMINISTRATOR DEFENSE TECHNICAL INFO CENTER ATTN: DTIC-FDAC CAMERON STATION ALEXANDRIA, VA 22304-6145	12	DIRECTOR US ARMY INDUSTRIAL BASE ENGR ACTV ATTN: AMXIB-P ROCK ISLAND, IL 61299-7260	1
COMMANDER US ARMY ARDEC ATTN: SMCAR-AEE	1	COMMANDER US ARMY TANK-AUTMV R&D COMMAND ATTN: AMSTA-DDL (TECH LIB) WARREN, MI 48397-5000	1
SMCAR-AES, BLDG. 321	1	COMMANDER US MILITARY ACADEMY	1
SMCAR-AET-O, BLDG. 351N	1	ATTN: DEPARTMENT OF MECHANICS	
SMCAR-CC	1	WEST POINT, NY 10996-1792	
SMCAR-CCP-A	1	US ARMY MISSILE COMMAND	
SMCAR-FSA	1	REDSTONE SCIENTIFIC INFO CTR	2
SMCAR-FSM-E	1	ATTN: DOCUMENTS SECT, BLDG. 4484	
SMCAR-FSS-D, BLDG. 94	1	REDSTONE ARSENAL, AL 35898-5241	
SMCAR-IMI-I (STINFO) BLDG. 59	2		
PICATINNY ARSENAL, NJ 07806-5000			
DIRECTOR US ARMY BALLISTIC RESEARCH LABORATORY ATTN: SLCBR-DD-T, BLDG. 305	1	COMMANDER US ARMY FGN SCIENCE AND TECH CTR ATTN: DRXST-SD	1
ABERDEEN PROVING GROUND, MD 21005-5066		220 7TH STREET, N.E. CHARLOTTESVILLE, VA 22901	
DIRECTOR US ARMY MATERIEL SYSTEMS ANALYSIS ACTV ATTN: AMXSY-MP	1	COMMANDER US ARMY LABCOM	
ABERDEEN PROVING GROUND, MD 21005-5071		MATERIALS TECHNOLOGY LAB ATTN: SLCMT-IML (TECH LIB)	2
COMMANDER HQ, AMCCOM ATTN: AMSMC-IMP-L	1	WATERTOWN, MA 02172-0001	
ROCK ISLAND, IL 61299-6000			

NOTE: PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER, US ARMY AMCCOM, ATTN: BENET LABORATORIES, SMCAR-CCB-TL, WATERVLIET, NY 12189-4050, OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST (CONT'D)

	<u>NO. OF COPIES</u>		<u>NO. OF COPIES</u>
COMMANDER US ARMY LABCOM, ISA ATTN: SLCIS-IM-TL 2800 POWDER MILL ROAD ADELPHI, MD 20783-1145	1	COMMANDER AIR FORCE ARMAMENT LABORATORY ATTN: AFATL/MN EGLIN AFB, FL 32542-5434	1
COMMANDER US ARMY RESEARCH OFFICE ATTN: CHIEF, IPO P.O. BOX 12211 RESEARCH TRIANGLE PARK, NC 27709-2211	1	COMMANDER AIR FORCE ARMAMENT LABORATORY ATTN: AFATL/MNF EGLIN AFB, FL 32542-5434	1
DIRECTOR US NAVAL RESEARCH LAB ATTN: MATERIALS SCI & TECH DIVISION CODE 26-27 (DOC LIB) WASHINGTON, D.C. 20375	1 1	MIAC/CINDAS PURDUE UNIVERSITY 2595 YEAGER ROAD WEST LAFAYETTE, IN 47905	1
DIRECTOR US ARMY BALLISTIC RESEARCH LABORATORY ATTN: SLCBR-IB-M (DR. BRUCE BURNS) ABERDEEN PROVING GROUND, MD 21005-5066	1		

NOTE: PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER, US ARMY AMCCOM, ATTN: BENET LABORATORIES, SMCAR-CCB-TL, WATERVLIET, NY 12189-4050, OF ANY ADDRESS CHANGES.