ssdnm
środowiskowe
studia doktoranckie
z nauk matematycznych

Grzegorz Jabłoński

Uniwersytet Jagielloński

Dimension of fractals

Praca semestralna nr 2

(semestr zimowy 2012/13)

Opiekun pracy: dr hab. Krzysztof Barański

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Dimension of fractals
Term paper

Grzegorz Jabłoński
Department of Mathematics and Computer Science
Uniwersystet Jagielloński, Kraków, Poland
grzegorz.jablonski@uj.edu.pl

January 16, 2013

1 Introduction

The term fractal describes specific family of sets in mathematics with two main characteristic features: self-similarity and irregularity [6]. The first approaches to the problem of fractals have been found in 17th century works of Leibniz. Then in 1883 George Cantor defined sets on real line, which have fractal properties. However, for a long time there was no much of development in that area until independent discoveries by Pierre Fatou and Gaston Julia in 1918 about functions in complex numbers.

In the last century, especially when computers started to be accessible for academic users, fractals were used in image compression [1], generating natural images (trees, clouds, mountains etc.), industry [5] as well as in medicine and biology [4] [8].

The simplest geometrical description of such complex structures is their fractal dimension. It is important not to use classical dimension because it was developed for less complicated sets.

The goal of this paper is to study the exactness of numerical approximations of dimension of fractals in case of known fractals (Iterated Function Systems and Julia sets). Further, we survey the state of the art methods to approximate dimension of fractals.

2 Hausdorff dimension

The Hausdorff dimension plays a central role in the description of fractals in mathematics. It is based on the Hausdorff measure which makes it mathematically easy to understand and analyse. Intuitively, the Hausdorff dimension describes how complicated the set is. Natural feature of sets in $\mathbb{R}^n$ is the *topological dimension*, which is 0 for totally disconnected sets, 1 if every point has neighbourhood with boundary of zero dimension and so forth. The topological dimension of sets in $\mathbb{R}^n$ agrees with our expectations that curves have dimension 1, triangles, circles and sets that can be drawn in $\mathbb{R}^2$ have dimension two. Such definition, however, does not contain information about more complex sets. Let us consider the *middle third Cantor set* - constructed by recursively removing open middle thirds of a parts of line - see fig. 1. Clearly its topological dimension is 0, because all points are isolated, but we can see it has more complicated structure.

Figure 1: Construction of middle third Cantor set (called also ternary set) - visible is result of 6 iterations.

To define more suitable dimension, we have to define the Hausdorff measure. First, we provide notation: *diameter* of a non-empty compact set $U \subset \mathbb{R}^n$ is $|U| := \sup\{|x-y| : x, y \in U\}$, σ*-cover* of F is a countable collection $\{U_i\}$ such that $|U_i| < \sigma$ and $F \subset \bigcup_{i=1}^{\infty} U_i$.

Given $F \subset \mathbb{R}^n$, let us write

$$\mathcal{H}_\delta^s(F) = \inf\{\sum_{i=1}^{\infty} |U_i|^s \colon \{U_i\} \text{ is a } \delta\text{-cover of } F\} \tag{1}$$

$\mathcal{H}_\delta^s(F)$ is the Hausdorff measure of F. We see that increasing δ we allow more covers. Thus when we decrease δ, $\mathcal{H}_\delta^s(F)$ is larger. Using this observation one can define *s-Hausdorff dimension*:

$$\mathcal{H}^s(F) = \lim_{\delta \to 0} \mathcal{H}_\delta^s(F) \tag{2}$$

$\mathcal{H}^s(F)$, by its construction fulfils two properties:

$$\mathcal{H}^s(F) = 0 \Rightarrow \forall_{t>s} H_t(F) = 0$$

and

$$\mathcal{H}^s(F) = \infty \Rightarrow \forall_{t<s} H_t(F) = \infty$$

Thus we define:

$$\dim_{\mathrm{H}} F = \sup\{s : \mathcal{H}^s(F) = \infty\} = \inf\{s : \mathcal{H}^s(F) = 0\} \tag{3}$$

It could be proven (see [6]) that $\mathcal{H}^s(F)$ is a measure and is non-increasing with s. Therefore, there exists s such that $\mathcal{H}^s$ changes from ∞ to 0. This value is called the *Hausdorff dimension* of F and will be denoted as $\dim_{\mathrm{H}} F$.

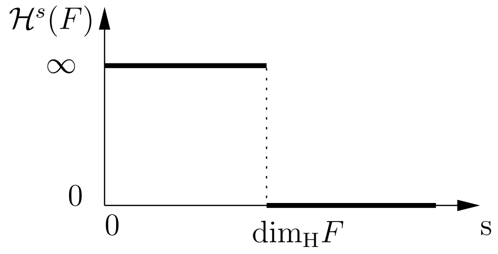

Figure 2: Graph of s-Hausdorff dimension and "jump" which is $\dim_H F$

3 Box counting dimension

The Hausdorff dimension has one obvious drawback - it is very hard to compute dimension for an arbitrary chosen set. This leads to invention of modifications of the Hausdorff dimension to make its computation more effective. One has to remember that such modifications alters crucial properties of the Hausdofrff dimension necessary to capture the information about complexity of the set. This is the trade-off between possibility to measure how complicated is set and easiness of computation.

There is a general rule how to build the dimension: given parameter δ we discard the features "smaller" than δ and measure the set. Such a measure is then compared in a logarithmic scale to δ.

The box counting dimension is an example of such approach. For a non-empty, bounded set $F \subset \mathbb{R}^n$ we count the minimal number of sets with diameter at most δ needed to cover F. Denote that number by $N_\delta(F)$. One can define the *upper* and *lower box counting dimension.*

$$\overline{\dim}_B F = \limsup_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta} \tag{4}$$

$$\underline{\dim}_B F = \liminf_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta} \tag{5}$$

If they agree, we call that value the *box counting dimension*:

$$\underline{\dim}_B F = \overline{\dim}_B F = \dim_B F \tag{6}$$

It is straightforward to observe that this definition allows to approximate the dimension of a given set by numerical experiments. One has to choose sufficiently small parameter δ, cover F with regular sets smaller than δ, for instance squares and count them.

The exact construction of the algorithm is presented in algorithm 1.

Algorithm 1 Box-counting method

1: **Input:** I - image of size n-by-n
2: **Output:** $\dim_B$ - approximated box counting dimension
3: let p be the maximal integer such that $2^p \leq n$
4: $G \leftarrow \emptyset$
5: **for** $\delta \leftarrow 1$ to p **do**
6: divide I into grid of size 2^δ
7: $N_\delta \leftarrow$ number of cells of grid that intersects with non-zero pixels of I
8: $G \leftarrow G \cup (-\log \delta, \log N_\delta)$
9: **end for**
10: fit the linear function f that describes points in G
11: return the slope of function f

We use the fact that box-counting dimension is defined as a limit with δ approaching zero, therefore in line 10 we try to fit the line that best describes the log-log graph of N_δ vs. δ. We can use least squares method or any other fitting method for linear functions. We will use this algorithm later in experiments in sections 4.2 and 5.2.

There is a general rule that states how box counting dimension and Hausdorff dimension of set F are ordered:

$$0 \leq \dim_{\mathrm{H}} F \leq \underline{\dim}_B F \tag{7}$$

Proof. We know from definition 1 that if F can be covered by $N_\delta(F)$ of sets of diameter at most δ then $\mathcal{H}^s_\delta(F)$ is the infimum of sum of diameters of all such sets, thus $\mathcal{H}^s_\delta(F) \leq N_\delta(F)\delta^s$. We can take $s < \dim_{\mathrm{H}} F$ and let $\delta \to 0$ such that

$$\lim_{\delta \to 0} \mathcal{H}^s_\delta(F) = \infty$$

then

$$0 \leq \log \mathcal{H}^s_\delta(F) \leq \log N_\delta(F)\delta^s = \log N_\delta(F) + s \log \delta$$

So

$$s \leq \underline{\inf}_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta}$$

□

4 Iterated Function Systems

Iterated functions systems are among one of the most popular fractals. Important properties of IFS are: self-similarity and simple construction. We have already showed one example of IFS, namely the Middle Third Cantor Set. Let us now take a closer look at IFS.

First, we recall some notation: a *contraction* is a mapping on $D \subset \mathbb{R}^n$ (D is closed), if there exists $c < 1$ such that $|S(x) - S(y)| \leq c|x - y|$. A finite family of contractions $\{S_1, S_2, \ldots, S_m\}$ is called an *Iterated Function System*. Important for us is the fact that IFS has its *attractor* F which is the set invariant with respect to the family of S_i:

$$F = \bigcup_{i=1}^{m} S_i(F) \tag{8}$$

Let us now present a fundamental theorem for iterated function systems:

Theorem 1. *For $D \subset \mathbb{R}^n$ and a family of contractions $S_1, S_2, \ldots, S_m$ there exists an uniquely determined non-empty compact attractor F such that* (8) *is fulfilled. The formula for F is:*

$$F = \bigcap_{k=0}^{\infty} S^k(E) \tag{9}$$

where $S(E) = \bigcup_{i=1}^{m} S_i(E)$, S^k is k-th iterate of S and E is a compact set such that $S_i(E) \subset E$.

The proof of the theorem can be found in [6].

The theorem gives us a method to approximate the attractor, which we call fractal. As a first step, we take any compact set, for instance unit square. Then we apply S iteratively. It could be shown that S^k converges to F.

$$d(S^k(E), F) \to 0 \text{ when } k \to \infty \tag{10}$$

Proof. Every S_i is a contraction mapping, so $d(S(A), S(B)) \leq Cd(A, B)$, where C is maximal among all c_i - values in definition of S_i. From the definition of F we have $d(S(E), F) = d(S(E), S(F))$. Putting all together we have: $d(S^k(E), F) = d(S^k(E), S^k(F)) \leq C^k d(A, B)$. (10) follows now immediately. □

The second method called *chaos game* is based on pointwise generation. We choose a point x_0, and apply randomly one of the contractions, say S_i to it: $x_1 = S_i(x_0)$. Then we proceed similarly to obtain more points: $x_k = S_{i_k}(x_{k-1})$, where S_{i_k} is again selected randomly. If we iterate long enough, the sequence of points will looks like the attractor F. We do not provide the mathematical proof that this algorithm works, but it is a consequence of ergodic theory. Result of such operation for Sierpiński triangle is visible in figure 4.

4.1 Hausdorff dimension

We are interested in iterated function system not only because of their straightforward computation but also the property of some class of IFS. We say that a family S_i fulfils the *open set condition*, if there exists non-empty bounded open set V such that

$$\bigcup_{i=1}^{m} S_i(V) \subset V \tag{11}$$

Such families have special properties, which allow to relate their Hausdorff dimension and box counting dimension:

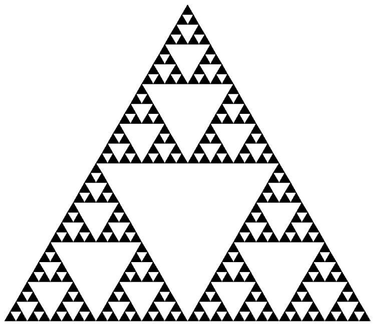

Figure 3: The Sierpiński triangle generated with five iterations of IFS

Theorem 2. *For an iterated function system $\{S_i\}$ fulfilling the open set condition, the Hausdorff and box dimension are equal to the value s such that the following equality holds*

$$\sum_{i=1}^{m} c_i^s = 1$$

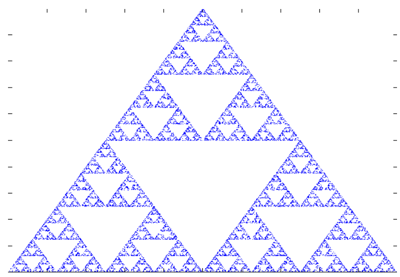

Figure 4: Sierpiński triangle generated using chaos game. We draw sequence of 20000 of points.

4.2 Experiment

We are now ready to perform first experiment. We generate the Sierpiński triangle (see figure 3) with the first method, and then approximate its dimension using box counting method.

Sierpiński triangle. The Iterated function system producing this figure consists of three contracting similarities that shrinks the triangle into three identical triangles but in different positions.

The IFS used to generate Sierpiński triangle has its Hausdorff dimension equal to box counting dimension. This follows from the fact that contraction fulfils the open set condition and theorem 2. Additionally, we can calculate $\dim_{\mathrm{H}} F$ by solving the equation $\sum_{i=1}^{3}(\frac{1}{2})^s = 1$ for s. A simple calculation gives us that $s = \frac{\log 2}{\log 3} \approx 1.585$.

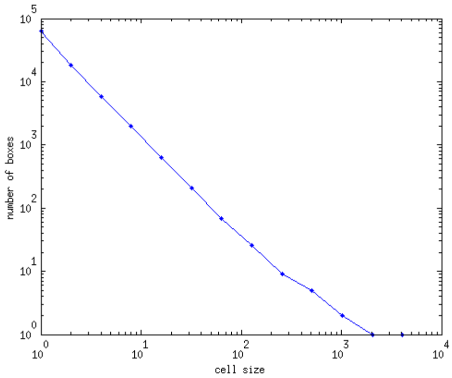

Figure 5: Log-log plot of δ vs. N_δ for the Sierpiński triangle. On X-axis we have the cell size and on Y-axis the number of boxes

In Figure 5 we show results of box-counting method for different cell sizes on the log-log plot. We generated the Sierpiński gasket with 5 iterations. The resulting box-counting dimension was obtained using the least-squares method to fit linear function to the graph. The value was 1.5064 which is in our opinion very close estimate for s.

5 Julia sets

Julia sets are among one of the most recognizable mathematical structures due to their extraordinary shapes. The study of Julia set and Fatou set which are complementary started with work by G. Julia [7] and P. Fatou in the twenties of 20th century, but they become very known when computers started to be more popular (see for instance [9]). For a comprehensive study of Julia set we refer reader to[11].

We present the definition of Julia set in terms of one variable complex analysis: Let f be a non-constant holomorphic mapping on the Riemann sphere $(\mathbb{C} \cup \infty)$. For every $z_0 \in S$, if there does not exist a neighbourhood of z_0 such that the sequence of compositions f^n is a normal family, then z_0 is in *Julia set*. In other words the Julia set consists of all points such that dynamics of f is strange in their neighbourhood.

We will restrict ourselves to a small family of Julia sets called J_c, that comes from quadratic mappings of form $f_c(z) = z^2 + c$ where c is fixed complex value. The reason is that we know the Hausdorff dimension of Julia set for small c. Moreover, we have methods to generate Julia sets, which we present in section 5.1.

5.1 Computer methods

Let us denote the n-th preimage of z as:

$$J_c^{-n}(z) = \{\bar{z} \mid J_c^n(\bar{z}) = z\}$$

First, we would like to have approximate image of the Julia set. Let us state the following theorem:

Theorem 3. *For every point $z \in J_c$ the set of preimages*

$$\bigcup_{n \geq 0} J_c^{-n}(z)$$

is everywhere dense in J_c

The proof of the theorem can be found in [11].

A straightforward algorithm based on the theorem is to take $z_0 \in J_c$ and then find preimages of z_0, then find the preimage of all points from $J_c^{-1}(z_0)$ and so forth. To compute preimage of point z_0 we solve the equation $J_c(z) = z_0$ for z, which in case of quadratic function is trivial $z_k = \sqrt{z_{k-1} - c}$. In order to get to all areas of the fractal we modify the method and choose only those solutions that guarantee going to not yet visited areas. The result of inverse method without modification is shown in fig 6. For more advanced methods of generating Julia sets we refer to [13].

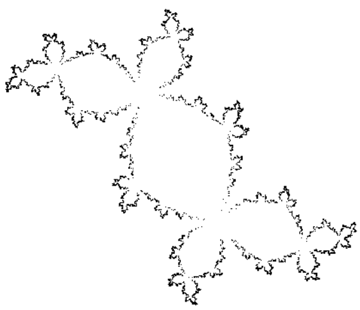

Figure 6: Julia set generated with inverse method for $c = -0.12 + 0.74i$, called also the Douady rabbit

We use three classes of Julia sets, that have known Hausdorff dimension. In order to test the box-counting method we compare obtained results with the exact value [10].

The first class is connected with a theorem proved by Ruelle [12]. For small c the following is true:

$$\dim_{\mathrm{H}}(J_c) = 1 + \frac{|c|^2}{4 \ln 2} + O(|c|^3)$$

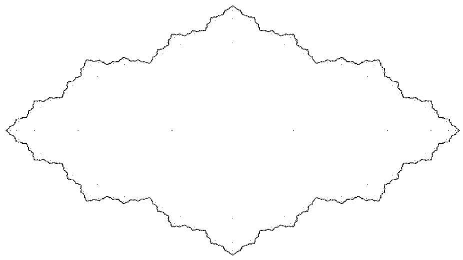

Figure 7: Julia set for $c = c_3 = -0.5$.

We will use $c = c_1 = -0.3$ for the first test, called later J_{c_1}

The second example is Julia set for $c = c_2 = -0.12+0.74i$ called the *Douady rabbit*. Its Hausdorff dimension was calculated and is 1.3934... . Its graphical approximation obtained with modified inverse method is plotted in figure 8.

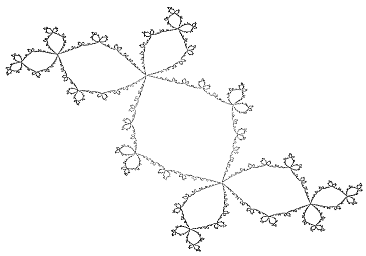

Figure 8: Douady rabbit generated with modified inverse method, $c = c_2 = -0.12 + 0.74i$. We can see that the result is more precise than for standard inverse method.

The last special case is Julia set for $c = c_3 = -1$, for which Hausdorff dimension is known to be 1.2683... . We show the visual representation in figure 9.

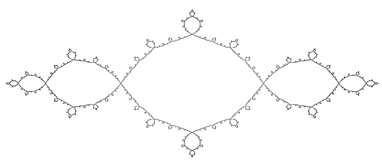

Figure 9: Julia set for $c = c_3 = -1$.

	J_{c_1}	J_{c_2}	J_{c_3}
True $\dim_{\mathrm{H}}$	1.0902	1.3934	1.2683
calculated $\dim_B$	1.0853	1.3440	1.2421

Table 1: Calculated $\dim_B$ for selected quadratic mappings.

5.2 Experiments

In this section we would like to show how exact BCM might be in case of Julia sets. We generated images of J_{c_1}, J_{c_2} and J_{c_3} using the FRACTINT application [14]. Every image was of size $1680x1240$ and generated using modified inverse method. Then we loaded the images to Matlab and computed the box counting dimension using algorithm 1. To fit the function we have used the linear least squares method. The calculation time did not exceeded one second in every case.

Results of our experiments are collected in the table 1. Clearly, the data shows that box-counting method gives close estimation. The second observation is that the calculated box-counting dimension is too small, however the difference is not substantial (error is always less than 4%). We should also point out that we used a very basic method without any optimization.

6 State of art methods

We now survey the development in the are of computing box counting method.

6.1 Differential box-counting method

Chaudhuri and Sarkar [3] proposed a method to compute dimension of gray level images. It means that in every pixel we have the intensity level of gray, instead of just binary information.

We proceed as in algorithm 1, but for every box we find the minimal (g_{min}) and maximal (g_{max}) value of all pixels inside. Then $N_r = g_{max} - g_{min} + 1$, and we estimate the dimension by finding the slope of linear approximation of log-log graph, as earlier.

There is a number of modifications of the above method, which looks to work for different problems including analysis of medical images [4] [8] or detection of defective fabrics in manufacture [5].

6.2 Modified box-counting method

Buczkowski et al. [2] proposed small modification to standard box-counting method, which as they state eliminate some of the problems connected with computation of BCM.

Their changes include the flexible way of setting the mesh size parameter. It is computed not for powers of two, but also for some interleaving values. As a consequence, we have more points to approximate the final result.

7 Concluding remarks

We show some basic examples of fractals like the Sierpiński triangle. Next, we presented a bit more complicated shapes such as the Julia set. We used the box-counting method to prove that it is useful in case of approximation of $\dim_B$. However, one has to remember that the Hausdorff dimension is not always the same as the box counting dimension. For some families of fractals the equality always holds. In practical considerations, as in cited articles about applications, one assumes that it is always true. The state of art methods show areas in which the application of fractal dimension might be useful. They also point out the direction of the development of modern computer methods for fractals.

References

[1] M. Barnsley and L. Barnsley. Fractal image compression. In *Image processing: mathematical methods and applications (Cranfield, 1994)*, Inst. Math. Appl. Conf. Ser. New Ser. Oxford Univ. Press, 1997.

[2] Stphane Buczkowski, Soula Kyriacos, Fahima Nekka, and LOuis Cartilier. The modified box-counting method: Analysis of some characteristic parameters. *Pattern Recognition*, 1998.

[3] B.B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1995.

[4] Toru Chikui, Kenji Tokumori, Kazunori Yoshiura, Kazunari Oobu, Seiji Nakamura, and Katsumasa Nakamura. Sonographic texture characterization of salivary gland tumors by fractal analyses. *Ultrasound in Medicine & Biology*, 2005.

[5] Aura Conci and Claudia Belmiro Proena. A fractal image analysis system for fabric inspection based on a box-counting method. *Computer Networks and ISDN Systems*, 1998.

[6] Kenneth Falconer. *Fractal Geometry Mathematical Foundations and Applications*. John & Wiley and Sons Inc., 2003.

[7] Gaston Julia. Mémoire sur l'iteration des fonctions rationnelles. *Journal de Mathmatiques Pures et Appliques*, 1918.

[8] Shoji Kido, Keiko Kuriyama, Masahiko Higashiyama, Tsutomu Kasugai, and Kuroda Chikazumi. Fractal analysis of small peripheral pulmonary nodules in thin-section ct: evaluation of the lung-nodule interfaces. *J Comput Assist Tomogr*, 2002.

[9] Benoit B. Mandelbrot. *The fractal geometry of nature*. W. H. Freeman and Co., 1982.

[10] Curtis McMullen. Hausdorff dimension and conformal dynamics, iii: Computation of dimension. *American Journal of Mathematics*, 2003.

[11] John Milnor. *Dynamics in one complex variable*. Annals of Mathematics Studies. Princeton University Press, 2006.

[12] David Ruelle. Repellers for real analytic maps. *Ergodic Theory Dynamical Systems*, 1982.

[13] Dietmar Saupe. Efficient computation of Julia sets and their fractal dimension. *Physica 28D*, 1987.

[14] Timothy Wegner and Jonathan Osuch. Fractint. `http://www.fractint.org/`, 2012.