A new approximation procedure for fractals

E. de Amo[a,*], I. Chiţescu[b], M. Díaz Carrillo[c], N.A. Secelean[d]

[a]*Departamento de Álgebra y Análisis Matemático, Universidad de Almería, 04120 Almería, Spain*
[b]*Facultatea de Matematica, Universitatea Bucureşti, Romania*
[c]*Departamento de Análisis Matemático, Universidad de Granada, Spain*
[d]*Department of Mathematics, University "Lucian Blaga" of Sibiu, Romania*

Abstract

This paper is based upon Hutchinson's theory of generating fractals as fixed points of a finite set of contractions, when considering this finite set of contractions as a contractive set-valued map.

We approximate the fractal using some preselected parameters and we obtain formulae describing the "distance" between the "exact fractal" and the "approximate fractal" in terms of the preselected parameters. Some examples and also computation programs are given, showing how our procedure works.
© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Hausdorff–Pompeiu distance; Contraction; Fixed point; Attractor; Fractal; Approximation procedure

1. Notations and preliminary facts

In order to be self-contained, we introduce notations and conventions (see [1,4]).

As usual, $\mathbb{R}$ = the reals, $\mathbb{R}_+$ = the positive reals, $\mathbb{R}_+^* = \mathbb{R} \setminus \{0\}$, $\mathbb{N}$ = the naturals. For every non-empty set X, $\mathrm{id}_X : X \to X$ is the identity of X, defined via $\mathrm{id}_X(a) = a$ for all a in X.

Considering a function $f : X \to X$, one can define $f^n : X \to X$ for all natural n, namely: $f^0 := \mathrm{id}_X$ and $f^n := f \circ f \circ \cdots \circ f$ (n-times composition) in case n is strictly positive. For such f we write $\mathrm{fix}(f)$ to denote the set of fixed points of f, i.e.

$$\mathrm{fix}(f) = \{x \in X \mid f(x) = x\}$$

(of course, it is possible to have $\mathrm{fix}(f) = \emptyset$ = the empty set).

---

* Corresponding author.

E-mail addresses: edeamo@ual.es (E. de Amo); madiaz@ugr.es (M. Díaz Carillo); secelean@mail.ulbsibiu.ro (N.A. Secelean).

If A is a subset of some topological space, $\bar{A}=$ the closure of A.

In the sequel, (X, d) will be a *complete metric space*. Write

$$\mathscr{K}(X) := \{A \mid \emptyset \neq A \subset X, \ A \text{ compact}\}.$$

If $x \in X$ and $\emptyset \neq A \subset X$, the distance between x and A (induced by d) will be

$$d(x, A) := \inf\{d(x, a) \mid a \in A\}$$

and, of course, in case $A = \{y\}$, one has $d(x, A) = d(x, y)$. One has $d(x, A) = 0$ if and only if $x \in \bar{A}$ and, in case $A \in \mathscr{K}(X)$, for every x in X, there exists a in A (depending upon x) such that $d(x, A) = d(x, a)$.

Now, let us consider $\emptyset \neq A \subset X$ and $r \in \mathbb{R}_+^*$. The open set

$$B(A, r) := \{x \in X \mid d(x, A) < r\}$$

is called *the open ball centered at A of radius r* (induced by d). Of course, in case $A = \{y\}$, $B(A, r) =: B(y, r) =$ the usual open ball centered at y of radius r. Similarly, the closed set

$$B[A, r] := \{x \in X \mid d(x, A) \leqslant r\}$$

called *the closed ball centered at A of radius r*, generalizes, in case $A = \{y\}$, the usual closed ball centered at y of radius r, namely one has, in this case, $B[A, r] =: B[y, r]$.

Now, let us consider A, B in $\mathscr{K}(X)$. The *distance between A and B* will be defined as follows:

$$D(A, B) := \inf\{r \mid r \in \mathbb{R}_+^*, \ A \subset B(B, r) \text{ and } B \subset B(A, r)\}.$$

One can prove the equality

$$D(A, B) := \inf\{r \mid r \in \mathbb{R}_+^*, \ A \subset B[B, r] \text{ and } B \subset B[A, r]\}.$$

It follows immediately that, in case $B \subset B' \subset A$, one has $D(A, B) \geqslant D(A, B')$.

It is very well known that $(\mathscr{K}(X), D)$ is *a complete metric space* (D is called the *Hausdorff–Pompeiu metric*). There is still another way of defining the metric D. Namely, for the some A, B in $\mathscr{K}(X)$, put

$$\rho(A, B) := \sup\{d(b, A) \mid b \in B\}.$$

Then, it is possible to prove the equality

$$D(A, B) = \max(\rho(A, B), \rho(B, A)).$$

For every $A \in \mathscr{K}(X)$, the *diameter of A* is the number

$$\mathrm{diam}(A) := \sup\{d(x, y) \mid x, y \in A\}.$$

Notice (do not forget that A is compact) the existence of x_0, y_0 in A such that $\mathrm{diam}(A) = d(x_0, y_0)$.

Recall that a function $f : X \to X$ is called a *contraction* in case there exists $0 < \lambda < 1$ such that, for all x, y in X,

$$d(f(x), f(y)) \leqslant \lambda d(x, y).$$

We shall say that λ is a *Lipschitz constant* for f. One can see that the infimum of all such λ, which will be denoted by $L(f)$ and will be called the *ratio of the contraction f*, also has the

property that, for all x, y in X,

$$d(f(x), f(y)) \leqslant L(f) d(x, y).$$

If $f_i : X \to X$, $i = 1, 2, \ldots, m$, are contractions such that, for all i and all x, y in X, one has

$$d(f_i(x), f_i(y)) \leqslant \lambda_i d(x, y)$$

then, writing $F = f_1 \circ f_2 \circ \cdots \circ f_m$, one has, for all x, y in X,

$$d(F(x), F(y)) \leqslant \lambda_1 \lambda_2 \ldots \lambda_m d(x, y)$$

which implies $L(f) = L(f_1) L(f_2) \ldots L(f_m)$.

The basic result of the construction is the *Banach–Caccioppoli–Picard contraction principle*. Let $f : X \to X$ be a contraction. Then, there exists an unique fixed point x^* of f. Moreover, x^* can be obtained as follows:

(a) take an arbitrary $x_0 \in X$;
(b) $x^* = \lim_n f^n(x_0)$.

Consequently, one has

Lemma 1. *Let $f : X \to X$ be a contraction and $\eta \in \mathbb{R}_+^*$. Then, one can find $y \in X$ such that $d(f(y), y) < \eta$.*

Proof. Let $0 < \lambda < 1$ such that $d(f(x), f(y)) \leqslant \lambda d(x, y)$ for all x, y in X. Take arbitrarily x_0 in X and write $x_{n+1} := f(x_n)$ for all natural n. So, $x_n \xrightarrow{n} x^*$, as we have seen. For $n \in \mathbb{N}$:

$$d(x_{n+1}, x_n) = d(f(x_n), f(x_{n-1})) \leqslant \lambda d(x_n, x_{n-1})$$

and, iterating

$$d(f(x_n), x_n) = d(x_{n+1}, x_n) \leqslant \lambda^n d(x_1, x_0).$$

For large enough n, one has $\lambda^n d(x_1, x_0) < \eta$. Write $y := x_n$ for such n.
Then

$$d(f(y), y) \leqslant \lambda^n d(f(x_0), x_0) < \eta. \qquad \square \tag{1}$$

Remark. Formula (1) which shows how to find practically y will be effectively used in Theorem 3 in the sequel.

From now on, we shall fix a natural number $m \geqslant 1$ and a finite set $\mathscr{S} = \{f_1, f_2, \ldots, f_m\}$ of contractions $f_i : X \to X$ with ratios r_i. These contractions define the functions $F : \mathscr{K}(X) \to \mathscr{K}(X)$ given by

$$F(C) := \bigcup_{i=1}^{m} f_i(C).$$

We finish this paragraph with the fundamental results which will be used in the rest of the paper.

Theorem 1. *The function F is a contraction on the metric space $(\mathcal{K}(X), D)$ with ratio*

$$L(F) \leqslant \max_{1 \leqslant i \leqslant m} r_i.$$

Using the preceding result and the contraction principle, we obtain the fundamental result of Hutchinson [5].

Theorem 2. *There exists an unique $A \in \mathcal{K}(X)$ such that $F(A) = A$. The set A is called the attractor of the family $\mathcal{S}$.*
Moreover, for every $B \in \mathcal{K}(X)$, the sequence $(F^n(B))_n$ converges to A in the space $(\mathcal{K}(X), D)$.

Remark. In case $m = 1$, taking $B = \{x\}$ for some $x \in X$, one obtain $A = \{x^*\} =$ the fixed point of f_1.

Consequently, for the sake of non-triviality, only the case $m \geqslant 2$ will be taken into consideration. In this case A is a "fractal" (in many situations). In order to continue, we shall write, for all natural $n \geqslant 1$, as follows: $\mathcal{S}_n :=$ the set of all possible compositions of the form $f_{i_1} \circ f_{i_2} \circ \cdots \circ f_{i_n}$, where $i_1, i_2, \ldots, i_n \in \{1, 2, \ldots, m\}$.
It is clear that $\mathcal{S}_n$ has at most m^n elements.
Of course, $\mathcal{S}_1 = \mathcal{S}$. Write also:

$$\mathcal{S}^* := \bigcup_{n=1}^{\infty} \mathcal{S}_n.$$

It is clear that, for all $B \in \mathcal{K}(X)$, one has

$$F^n(B) = \bigcup_{f \in \mathcal{S}_n} f(B)$$

and $F^{n+1}(B) = F(F^n(B))$.
Another result of Hutchinson is the following: If A is the attractor of $\mathcal{S}$, then

$$A = \overline{\bigcup_{f \in \mathcal{S}^*} \operatorname{fix}(f)}$$

which implies that, for all natural $n \geqslant 1$ and for all $f \in \mathcal{S}_n$, one has $\operatorname{fix}(f) \subset A$ (of course, all $\operatorname{fix}(f)$ are singletons).

Lemma 2. *Let A be the attractor of $\mathcal{S}$. Then, for every natural $N \geqslant 1$, A is the attractor of $\mathcal{S}_N$.*

Proof. Induction upon N. For $N = 1$, the result is true according to the definition.
Now, accept that A is the attractor of $\mathcal{S}_{N-1}$, i.e.

$$A = \bigcup_{i_1, \ldots, i_{N-1}} f_{i_1} \circ \cdots \circ f_{i_{N-1}}(A)$$

the union being performed over all possible $i_1, i_2, \ldots, i_{N-1}$ in $\{1, 2, \ldots, m\}$.

Consequently,

$$A = \bigcup_{j=1}^{m} f_j(A) = \bigcup_{j,i_1,\dots,i_{N-1}} f_j \circ f_{i_1} \circ \cdots \circ f_{i_{N-1}}(A) = \bigcup_{f \in \mathscr{S}_N} f(A). \qquad \square$$

2. A new procedure of approximating fractals

With the help of Iterated Function Systems, we improve the known techniques for approximation of fractals (see [2,3]). We use the notations in the preceding paragraph, with the same meanings: (X, d), $\mathscr{S} = \{f_1, f_2, \dots, f_m\}$, $m \geqslant 2$, contractions with Lipschitz constants (or ratios) $r_1, r_2, \dots, r_m$.

For a given number $\varepsilon \in \mathbb{R}_+^*$, one can find a natural number $N \geqslant 1$ such that

$$r_{i_1} r_{i_2} \dots r_{i_N} \leqslant \varepsilon$$

for all possible choices of $i_1, i_2, \dots, i_N$ in $\{1, 2, \dots, m\}$. Call the set of all such numbers N as follows: $A(\mathscr{S}, \varepsilon)$. So, there exists a natural $N_0 \geqslant 1$ such that $A(\mathscr{S}, \varepsilon) = \{N_0, N_0 + 1, N_0 + 2, \dots\}$.

Let us pass to our main results.

Theorem 3. *Let the numbers* $1 > \delta > 0$, $1 > \varepsilon > 0$ *and* $\eta \geqslant 0$ *be given. Let* $N \in A(\mathscr{S}, \delta)$ *and consider the set* $\mathscr{S}_N = \{h_1, h_2, \dots, h_u\}$ *(of course* $u \leqslant m^N$*).*

One can find, for every $i = 1, 2, \dots, u$, *an element* $x_i \in X$ *such that* $d(x_i, h_i(x_i)) \leqslant \eta$. *Denote by* B_1 *the finite set of all such* x_i.

Take $M \in A(\mathscr{S}, \varepsilon)$ *and put*

$$B = \{g(x) \mid g \in \mathscr{S}_M, \ x \in B_1\}.$$

Then, if A *is the attractor of* $\mathscr{S}$, *one has*

$$D(A, B) \leqslant \varepsilon \delta \operatorname{diam}(A) + \frac{\varepsilon \eta}{1 - \delta}.$$

Proof. Firstly, let us explain how the elements x_i occur. In case $\eta = 0$, x_i must be the unique fixed point of the contraction h_i. In case $\eta > 0$, one can apply Lemma 1 (and x_i *need not be* the fixed point of h_i). Use effectively (1).

We have

$$B = \bigcup_{g \in \mathscr{S}_M} g(B_1) = F^M(B_1) \quad \text{and} \quad A = F^M(A)$$

(according to Lemma 2). Hence, using Theorem 1, one can have

$$D(A, B) \leqslant L(F^M) D(A, B_1) \leqslant \varepsilon D(A, B_1) \tag{2}$$

(use the definition of M).

The rest of the proof will be divided into two steps.

First step:

$$\rho(A, B_1) = \max_{1 \leqslant i \leqslant u} d(x_i, A). \tag{3}$$

Take some $x_i \in B_1$ and let a_i be the fixed point of h_i. Then

$$d(x_i, a_i) = d(x_i, h_i(a_i)) \leqslant d(x_i, h_i(x_i)) + d(h_i(x_i), h_i(a_i)) \leqslant \eta + \delta d(x_i, a_i)$$

hence

$$d(x_i, a_i) \leqslant \frac{\eta}{1 - \delta}. \tag{4}$$

Because $a_i \in A$ (as we have already noticed), one obtains $d(x_i, A) \leqslant \eta/(1 - \delta)$ hence, using (3),

$$\rho(A, B_1) \leqslant \frac{\eta}{1 - \delta}. \tag{5}$$

Second Step:

$$\rho(B_1, A) = \sup\{d(a, B_1) \mid a \in A\}. \tag{6}$$

Take some $a \in A = F^N(A) = \bigcup_{i=1}^{u} h_i(A)$, and let $1 \leqslant i \leqslant u$, $z \in A$, such that $a = h_i(z)$. According to (4), one has (denoting again $a_i :=$ the fixed point of h_i)

$$d(a, B_1) \leqslant d(a, x_i) = d(h_i(z), x_i) \leqslant d(h_i(z), h_i(x_i)) + d(h_i(x_i), x_i) \leqslant \delta d(z, x_i) + \eta$$

$$\leqslant \delta(d(z, a_i) + d(a_i, x_i)) + \eta \leqslant \delta(\mathrm{diam}(A) + \eta/(1 - \delta)) + \eta$$

$$= \delta \, \mathrm{diam}(A) + \eta/(1 - \delta).$$

According to (6), the last inequality gives

$$\rho(B_1, A) \leqslant \delta \, \mathrm{diam}(A) + \frac{\eta}{1 - \delta}. \tag{7}$$

From (5) and (7), one obtains

$$D(A, B_1) \leqslant \delta \, \mathrm{diam}(A) + \frac{\eta}{1 - \delta}$$

and (2) finally gives

$$D(A, B) \leqslant \varepsilon \delta \, \mathrm{diam}(A) + \frac{\varepsilon \eta}{1 - \delta}. \qquad \square$$

Theorem 4. *Assume the conditions in the preceding theorem are fulfilled and assume also that $2\varepsilon\delta < 1$.*
Then, one has

$$D(A, B) \leqslant \frac{\varepsilon \delta}{1 - 2\varepsilon\delta} \, \mathrm{diam}(B) + \frac{\varepsilon \eta}{(1 - \delta)(1 - 2\varepsilon\delta)}.$$

Proof. We have already seen that

$$\rho(B, A) \leqslant D(A, B) \leqslant \varepsilon \delta \, \mathrm{diam}(A) + \frac{\varepsilon \eta}{1 - \delta}. \tag{8}$$

Let y_1 and y_2 be arbitrarily taken in A. Then

$$d(y_i, B) \leqslant \rho(B, A), \quad i = 1, 2. \tag{9}$$

One can find $b_i \in B$ such that $d(y_i, B) = d(y_i, b_i)$, $i = 1, 2$.

So, for arbitrary y_i, y_2 in A, we could find, using (8) and (9), the point b_1, b_2 in B such that

$$d(y_i, b_i) \leqslant \varepsilon\delta \operatorname{diam}(A) + \frac{\varepsilon\eta}{1 - \delta}, \quad i = 1, 2.$$

Hence

$$d(y_1, y_2) \leqslant d(y_1, b_1) + d(b_1, b_2) + d(b_2, y_2) \leqslant 2\varepsilon\delta \operatorname{diam}(A) + \frac{2\varepsilon\eta}{1 - \delta} + \operatorname{diam}(B).$$

Because y_1 and y_2 are arbitrary, we get

$$\operatorname{diam}(A) \leqslant 2\varepsilon\delta \operatorname{diam}(A) + \frac{2\varepsilon\eta}{1 - \delta} + \operatorname{diam}(B)$$

which implies

$$\operatorname{diam}(A) \leqslant \frac{\operatorname{diam}(B)}{1 - 2\varepsilon\delta} + \frac{2\varepsilon\eta}{(1 - \delta)(1 - 2\varepsilon\delta)}. \tag{10}$$

The last inequality exploited in (8) gives

$$D(A, B) \leqslant \varepsilon\delta \left[\frac{\operatorname{diam}(B)}{1 - 2\varepsilon\delta} + \frac{2\varepsilon\eta}{(1 - \delta)(1 - 2\varepsilon\delta)} \right] + \frac{\varepsilon\eta}{1 - \delta}$$

$$= \frac{\varepsilon\delta}{1 - 2\varepsilon\delta} \operatorname{diam}(B) + \frac{\varepsilon\eta}{(1 - \delta)(1 - 2\varepsilon\delta)}. \qquad \square$$

It is interesting and important to see what happens in the particular case when $\eta = 0$. Under this condition, $x_i = h_i(x_i)$ for all $h_i \in \mathscr{S}_N$. So, x_i are the fixed point of h_i, $i = 1, 2, \dots, u$. It follows that all x_i are in A, so $B_1 \subset A$, which in turn implies $B \subset A$ (indeed: $y \in B$ implies the existence of some $x \in B_1 \subset A$ and of some $g \in \mathscr{S}_M$ such that $y = g(x)$ and $A = F^M(A)$).

Summarizing the preceding considerations and using (10), we obtain

Theorem 5. *Let the numbers ε, δ be such that $1 > \delta > 0$, $1 > \varepsilon > 0$ and $2\varepsilon\delta < 1$. Let $N \in A(\mathscr{S}, \delta)$ and consider the set $\mathscr{S}_N = \{h_1, h_2, \dots, h_u\}$.*

For every $i = 1, 2, \dots, u$, let $x_i \in X$ be the unique fixed point of h_i and denote by B_1 the finite set of all these x_i.

Take $M \in A(\mathscr{S}, \varepsilon)$ and put

$$B = \{g(x) \mid g \in \mathscr{S}_M, \ x \in B_1\}.$$

Then, if A is the attractor of $\mathscr{S}$, one has

$$D(A,B) \leqslant \varepsilon\delta \operatorname{diam}(A) \leqslant \frac{\varepsilon\delta}{1-2\varepsilon\delta} \operatorname{diam}(B)$$

and $B \subset A$.

2.1. Comments

1. Our Theorems 3–5 give an estimation for the distance between the attractor (fractal) A and the approximating set B, in terms of preassigned parameters $\delta, \varepsilon, \eta$.
2. The set A generally cannot be "reached", whereas the finite set B which approximates A can be constructed after a finite number of steps (which can be very big in case δ, ε and η impose a very good approximation).
3. The use of the "parasite" number η is generally speaking necessary because the computation of the fixed points a_i of h_i can be very difficult or even impossible and this imposes the use of Lemma 1. So, we are obliged to use the "parasite" set B_1, which generates the approximating set B.
4. Theorem 3 gives an estimation of the distance between the attractor A (unknown) and the approximating set B (known = can be constructed) in terms of $\delta, \varepsilon, \eta$, but also in terms of A. This cannot be useful. So, Theorem 4 appears to be more useful, the estimation being only in terms of $\delta, \varepsilon, \eta$ and B.
5. The "ideal" situation is exposed in Theorem 5. Namely, in this case, only the "essential" approximation parameters appear. In this case the approximation is made "from inside", because $B \subset A$, so the approximating set B contains only points of the attractor A.
6. The particular case $N=1$ for Theorem 5 is also very important (i.e. we start with the set $B_1 = \{x_1, x_2, \ldots, x_m\}$ of fixed points of the functions $f_1, f_2, \ldots, f_m : f_i(x_i) = x_i$).

So, we can take $\delta := \max\{r_1, r_2, \ldots, r_m\}$ and $\varepsilon > 0$ such that $2\varepsilon\delta < 1$. Let $M \in A(\mathscr{S}, \varepsilon)$ (e.g. take M such that $\delta^M < \varepsilon$) and put

$$B = \{g(x) \mid g \in \mathscr{S}_m, \ x \in B_1\}.$$

Then, if A is the attractor of $\mathscr{S}$, one has $B \subset A$ and

$$D(A,B) \leqslant \varepsilon\delta \operatorname{diam}(A) \leqslant \frac{\varepsilon\delta}{1-2\varepsilon\delta} \operatorname{diam}(B). \tag{11}$$

In this case, after computing B_1, δ and M, the iterations will be

$$F(B_1) = V_1, \quad F(V_1) = V_2 = F^2(B_1), \ldots, V_m = F^M(B_1)$$

and finally one has $B = V_m$ and (11).

It is to be seen that $V_1 \subset V_2 \subset \cdots \subset V_m \subset A$ and $(D(A, V_n))_n$ is a decreasing sequence such that $\lim_n D(A, V_n) = 0$.

2.2. Practical application of the algorithm

In order to understand the underlying programs, we shall explain how Theorem 5 works, in the particular case of some attractors in $\mathbb{R}^2$ which are generated by affine contractions of the type

$$\omega(x, y) = (a_1 x + a_2 y + c_1, a_3 x + a_4 y + c_2)$$

with $a_1, a_2, a_3, a_4, c_1, c_2 \in \mathbb{R}$.

Let us assume that the m contractions $f_1, f_2, \ldots, f_m$ are given in matricial form via

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = A_i \begin{pmatrix} x \\ y \end{pmatrix} + C_i, \quad i = 1, 2, \ldots, m,$$

where

$$A_i = \begin{pmatrix} a_1(i) & a_2(i) \\ a_3(i) & a_4(i) \end{pmatrix}, \quad C_i = \begin{pmatrix} c_1(i) \\ c_2(i) \end{pmatrix}.$$

Then, for i, j in $\{1, 2, \ldots, m\}$, one has

$$f_i \circ f_j \begin{pmatrix} x \\ y \end{pmatrix} = A_i A_j \begin{pmatrix} x \\ y \end{pmatrix} + A_i C_j + C_i.$$

Hence, using an iterative process, one obtains the coefficients of the affine contractions in S_N. Afterwards, one computes the fixed points $(x, y) \in B_1$ and one continues the process until the computation of the coefficients of the elements in S_M. So, for $h_s \in S_N$:

$$h_s \begin{pmatrix} x \\ y \end{pmatrix} = AR_s \begin{pmatrix} x \\ y \end{pmatrix} + CR_s, \quad AR_s = \begin{pmatrix} ar_1(s) & ar_2(s) \\ ar_3(s) & ar_4(s) \end{pmatrix}, \quad CR_s = \begin{pmatrix} cr_1(s) \\ cr_2(s) \end{pmatrix}$$

and the fixed points will be

$$\begin{pmatrix} x(s) \\ y(s) \end{pmatrix} = CR_s(I - AR_s)^{-1}, \quad \text{where } I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

for $s = 1, 2, \ldots, m^N$.

One continues the computation and one represents the points $(p_1, p_2) \in B$ as images of the elements $(x(s), y(s), s = 1, 2, \ldots, m^N$, from B_1, when applying to them the functions in S_M.

In the sequel, one presents the Visual C program which uses the above described pattern followed by some examples of fractals in $\mathbb{R}^2$ obtained via this program, together with the respective parameters which have been used. For each fractal one gives also the time which is necessary for a Pentium III PC (600 MHz) to draw the respective image, the number of points that where computed and the number of distinct points that where drawn in each case. We lay stress upon the fact that "the factor of image dilation" (fa) represents an increasing factor for the image in order to permit a good visualisation of it on the display. To center the image on the display, one uses "the image center coordinates" (x0, y0).

2.3. *Visual C program*[1]

```
# include <math.h> float FAR *a1, FAR *a2, FAR *a3, FAR *a4, FAR*c1,
FAR*c2, FAR *ap1, FAR *ap2, FAR  *ap3, FAR *ap4, FAR *cp1, FAR*cp2,
FAR*ar1, FAR *ar2, FAR  *ar3, FAR *ar4, FAR *cr1, FAR *cr2, FAR  *x,
FAR  *y; char FAR  *tab; unsigned long aux, i,j,k,k1, l,m,n,s,N;
HANDLE h; LARGE_INTEGER lpPerfCounter_1; LARGE_INTEGER
lpPerfCounter_2; LARGE_INTEGER lpFrequency; long points,
distinct_points; SYSTEMTIME start_1, stop_1; double e; DWORD size
=2200000; int x0,y0,x1,y1; double
xmin,xmax,ymax,r,rr,delta,eps,fa,p1,p2,nr;
        h=HeapCreate(0,0,0);
        a1 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        a2 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        a3 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        a4 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        c1 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        c2 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        ap1 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        ap2 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        ap3 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        ap4 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        cp1 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        cp2 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        ar1 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        ar2 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        ar3 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        ar4 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        cr1 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        cr2 = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        x = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        y = (float FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
        tab = (char FAR*)HeapAlloc(h,HEAP_ZERO_MEMORY,size);
    QueryPerformanceFrequency(& lpFrequency);
    QueryPerformanceCounter (& lpPerfCounter_1);
i=1; rr=r; while(rr>=delta){rr*=r; i++; } k1=i; while (rr>=eps){
rr*=r; i++; } k=i; for (j=0;j<m;j++) { ar1[j]=1; ar2[j]0;
cr1[j]=0; ar3[j]=0; ar4[j]=1; cr2[j]=0; } l=1; for (n=1;n<=k;n++) { s=0;
for (j=0;j<1;j++) {ap1[j]=ar1[j]; ap2[j]=ar2[j];
ap3[j]=ar3[j]; ap4[j]=ar4[j]; cp1[j]=cr1[j]; cp2[j]=cr2[j]; }
for (j=0;j<1;j++) for (i=0;i<m;i++)
```

```
{ar1[s]=a1[i]*ap1[j]+a2[i]*ap3[j];
ar2[s]=a1[i]*ap2[j]+a2[i]*ap4[j];
ar3[s]=a3[i]*ap1[j]+a4[i]*ap3[j];
ar4[s]=a3[i]*ap2[j]+a4[i]*ap4[j];
cr1[s]=a1[i]*cp1[j]+a2[i]*cp2[j]+c1[i];
cr2[s]=a3[i]*cp1[j]+a4[i]*cp2[j]+c2[i]; s++;} if (n==k1)
for (s=0;s<l*m;s++){nr=((ar1[s]-1)*(ar4[s]-1)+(ar2[s]*ar3[s]));
x[s]=(cr1[s]*(1-ar4[s]+cr2[s]*ar2[s])/nr;
y[s]=(cr2[s]*(1-ar1[s]+cr1[s]*ar3[s])/nr;}l*=m;} n=1;
for (aux=1;aux<=k1;aux++) n*=m; l=n; for (aux=k1+1;aux<=k;aux++)
l*=m; points=0; for (j=0;j<l;j++) for (i=0;i<n;i++) {
p1=ar1[j]*x[i]+ar2[j]*y[i]+cr1[j]; if (xmin>p1) {xmin=p1; } else
if (xmax<p1) xmax=p1;p2=ar3[j]*x[i]+ar4[j]*y[i]+cr2[j];
if (ymin>p2){ymin=p2; } else if (ymax<p2) ymax=p2;
x1=(int)(floor(p1*fa)+x0); y1=(int)(floor(p2*fa)+y0);
tab[(x1*1000+y1)]=1; points++;}distinct_points=0;
for (i=0;i<1000;i++) for (j=0;j<700;j++) if (tab[(i*1000+j)])
{pdc->SetPixel(i,j,RGB(0, 0, 0)); distinct_points++; }
QueryPerformanceCounter(& lpPerfCounter_2); e=
eps*delta/(1-2*eps*delta)*
sqrt((xmax-xmin)*(xmax-xmin)+(ymax-ymin)*(ymax-ymin));HeapFree(h,0,(LPVOID)a1);
HeapFree (h, 0, (LPVOID)a2); HeapFree(h, 0,(LPVOID)a3);
HeapFree (h, 0, (LPVOID)a4); HeapFree(h, 0,(LPVOID)c1);
HeapFree (h, 0, (LPVOID)c2); HeapFree(h, 0,(LPVOID)ap1);
HeapFree (h, 0, (LPVOID)ap2); HeapFree(h, 0,(LPVOID)ap3);
HeapFree (h, 0, (LPVOID)ap4); HeapFree(h, 0,(LPVOID)cp1);
HeapFree (h, 0, (LPVOID)cp2); HeapFree(h, 0,(LPVOID)ar1);
HeapFree (h, 0, (LPVOID)ar2); HeapFree(h, 0,(LPVOID)ar3);
HeapFree (h, 0, (LPVOID)ar4); HeapFree(h, 0,(LPVOID)cr1);
HeapFree (h, 0, (LPVOID)cr2); HeapFree(h, 0,(LPVOID)x);
HeapFree (h, 0, (LPVOID)y); HeapFree(h, 0,(LPVOID)tab);
HeapDestroy(h);
CString str,str2; str2.Format(''Error %g\r\n'', e); str += str2; double fDif;
fDif=((double)(lpPerfCounter_2.QuadPart -
lpPerfCounter_1.QuadPart))/
((double)lpFrequency.QuadPart); str2.format (''Time in seconds is %.15f \r \n'',
fDif); str +=str2; str2.Format (''Number of Points %d \r\n Number of Distinct
Points %d'', points, distinct_points); str += str2; AfxMessageBox(str);
```

2.4. Examples
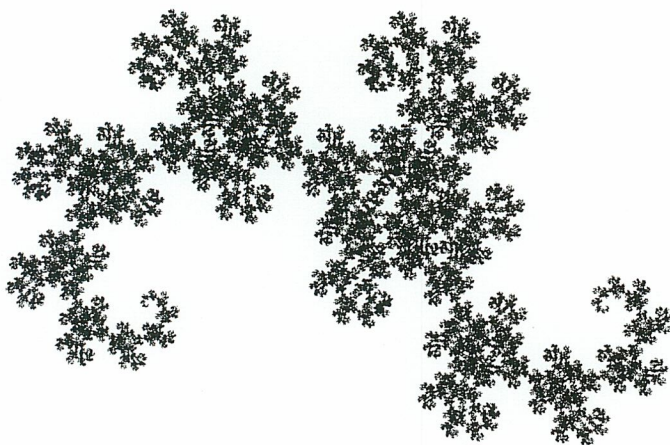
```
E1. Dragon
m=2;
r=sqrt(2)/2;
```

```
delta=eps=0.0626;
a1[0]=0.5;          a2[0]=0.5;          c1[0]=0;
a3[0]=-0.5;         a4[0]=0.5;          c2[0]=0;
a1[1]=-0.5;         a2[1]=0.5;          c1[1]=1;
a3[1]=-0.5;         a4[1]=-0.5;         c2[1]=0;
fa=600; x0=270; y0=410;
time=0.222 sec
number of points = 65536
number of distinct points = 57049
```
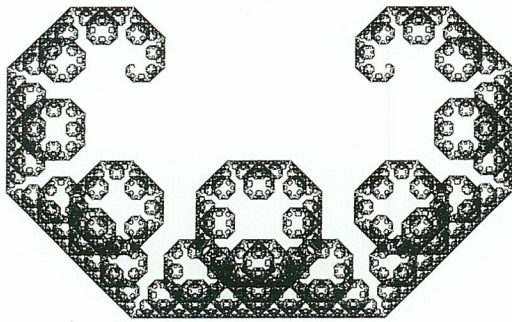

E2. Sierpinski gasket

```
m=3;
r=0.5;
delta=eps=0.0313;
a1[0]=0.5;          a2[0]=0;            c1[0]=0;
a3[0]=0;            a4[0]=0.5;          c2[0]=0;
a1[1]=0.5;          a2[1]=0;            c1[1]=0.25;
a3[1]=0;            a4[1]=0.5;          c2[1]=sqrt(3)/4;
a1[2]=0.5;          a2[2]=0;            c1[2]=0.5;
a3[2]=0;            a4[2]=0.5;          c2[2]=0;
fa=700;             x0=170;            y0=10;
time=0.187368 s
number of points = 59049
number of distinct points = 376557
```

```
E3. Levy curve

m=2;
r=sqrt(2)/2;
delta=eps=0.058;
a1[0]=0.5;          a2[0]=-0.5;        c1[0]=0;
a3[0]=0.5;          a4[0]=0.5;         c2[0]=0;
a1[1]=-0.5;         a2[1]=0.5;         c1[1]=0.5;
a3[1]=-0.5;         a4[1]= 0.5;        c2[1]=0.5;
fa=480;             x0=270;            y0=130;
time=0.623 sec
number of points = 262144
number of distinct points = 151442
```
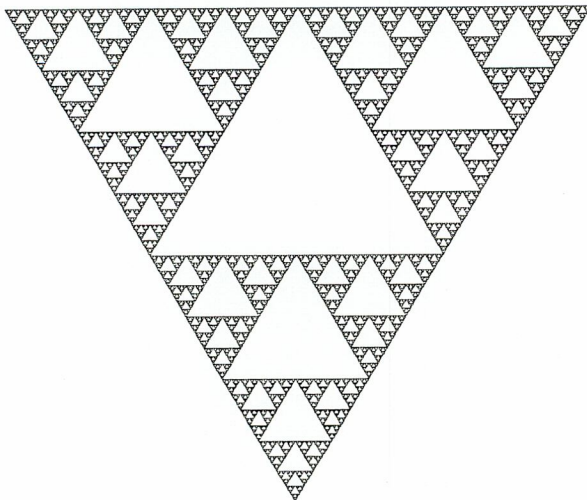
```
E4. Little Pine
m=3;
r=sqrt(2)/2;
delta=eps=0.2;
a1[0]=0.5;          a2[0]=-0.1;        c1[0]=0;
a3[0]=0.5;          a4[0]=0.5;         c2[0]=0;
a1[1]=-0.5;         a2[1]=-0.1;        c1[1]=-0.5;
a3[1]=-0.5;         a4[1]=0.6;         c2[1]=-2;
a1[2]=-0.7;         a2[2]=-0.3;        c1[2]=0;
a3[2]=-0.5;         a4[2]=0.5;         c2[2]=1;
fa=78;              x0=350;            y0=410;
time=0.144 sec
number of points = 59049
number of distinct points = 33536
```
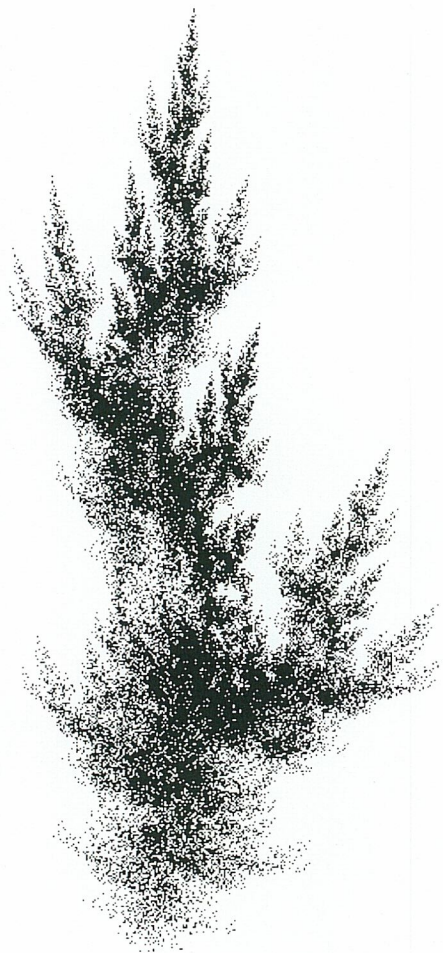
```
E5. Fine Dragon

m=2;
r=sqrt(2)/2;
delta=eps=0.0626;
a1[0]=-0.5;            a2[0]=-0.5;           c1[0]=1;
a3[0]=-0.333;         a4[0]=0.333;          c2[0]=3;
a1[1]=-0.5;           a2[1]=0.5;            c1[1]=-5;
a3[1]=-0.5;           a4[1]=-0.5;           c2[1]=-3;
fa=38;               x0=400;               y0=365;
time=0.133 sec
number of points = 65536
number of distinct points = 28979
```
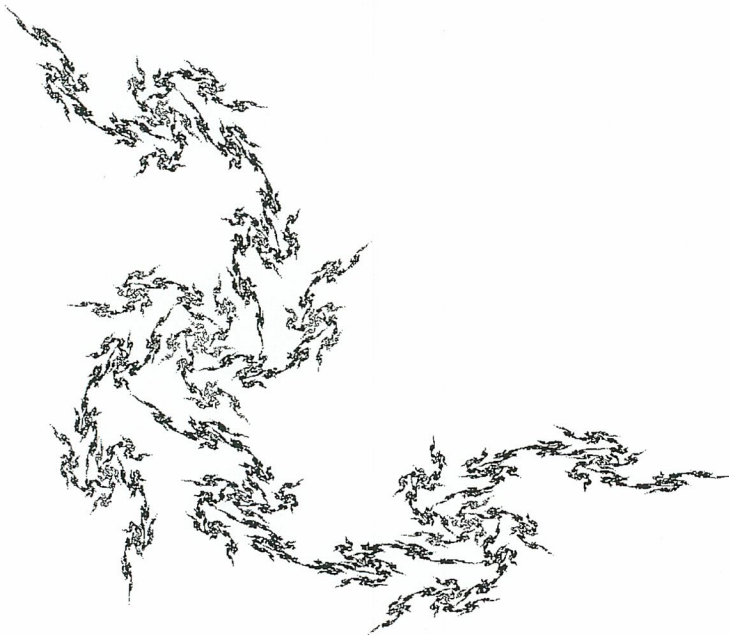

Acknowledgements

References

[1] M.F. Barnsley, Fractals Everywhere, Academic Press, Harcourt Brace Janovitch, New York, 1988.

[2] M.F. Barnsley, S. Demko, Iterated function system and the global construction of fractals, Proc. Roy. Soc. London Ser. A 399 (1985) 243–275.

[3] S. Dubuc, A. Elqortobi, Approximations of fractal sets, J. Comput. Appl. Math. 29 (1990) 79–84.

[4] K.J. Falconer, The Geometry of Fractal Sets, Cambridge University Press, Cambridge, 1985, p. 85.

[5] J. Hutchinson, Fractals and self-similarity, Indiana Univ. J. Math. 30 (1981) 713–747.