Estimation Correlation Dimension of the Filled Julia Set Generated by Escape Time Algorithm

1Arkan Jassim Mohammed and 2Noora Ahmed Mohammed

1,2Al-Mustansiryah University, College of Science, Department of Mathematics, Baghdad, Iraq.

Address For Correspondence:
Arkan Jassim Mohammed, Al-Mustansiryah University, College of Science, Department of Mathematics, Baghdad, Iraq.

A B S T R A C T

Fractal dimension is a measure of how fragmented of fractal object is. In fractal geometry, there are various approaches to compute the fractal dimension of an object. In this paper we propose method to compute correlation dimension of the Filled Julia set is obtained by using Grassberger-Procaccia algorithm, which is based on computing the correlation function. Computing the correlation function was implemented using the Matlab program. A log-log (logarithm scale) graph of the correlation function versus the distances between every pair of points in the fractal is an approximate of the correlation dimension of the fractal.

Keywords: Fractal dimension, Julia set, Filled Julia set, Correlation dimension, Escape Time Algorithm.

INTRODUCTION

Fractals are considered as significant tools in natural science, they are of great interest to graphics designers for their ability to simulate most of the natural phenomena. The term fractal is coined by Mandelbrot (1999), derived from the Latin adjective "fractus" which means broken or fragmented. Many objects in nature are very complex and irregular, therefore their description with traditional methods is insufficient. One possible approach is the fractal geometry. Fractal geometry is a useful way to describe and characterize complex shapes and surfaces. Fractal dimension $D$ is a key quantity in fractal geometry. The $D$ value can be a non-integer and can be used as an indicator of the complexity of the curves and surfaces. For a self-similar figure, it can be decomposed into N small parts, where each part is a reduced copy of the original figure by ratio $r$. The $D$ of a self-similar figure, it can be defined as $D = -\log(N) / \log(r)$. For curves and images that are not self-similar, there exist numerous empirical methods to compute $D$ results from different estimators were found to differ from each other. This is partly due to the elusive definition of $D$, i.e., the Hausdorff-Besicovitch dimension (Zhou, G., N.S.N. Lam, 2005). There are many kinds of dimensions. The correlation dimension is one of the fractal dimension measurement because it permits non-integer values. In (1983) Grassberger and Procaccia (Grassberger, P. and I. Procaccia, 1983a; Grassberger, P. and I. Procaccia, 1983b) published an algorithm (GPA) for estimating the correlation dimension that has become very popular and has been widely used since.

The correlation dimension $D_{cor}$ can be calculated in real time as the fractal generated by Escape Time Algorithm by using the distances between every pair of points in the attractor set of N number of points, it is so-called correlation function $C(\varepsilon)$. It is defined as the probability that two arbitrary points on the fractal are closer together than the sides of size $\varepsilon$ of the cells which cover the fractal. In this paper, we compute the correlation
dimension $D_{cor}$ for the Filled Julia set $F_c$, we will define two kinds of sets that are defined in terms of complex functions; filled Julia sets and Julia sets as an examples of fractal sets constructed by the Escape Time Algorithm.


Let $c$ be any complex number. The smallest closed set in the complex plane that contains all repelling periodic points of $f_c$ is called the Julia set of $f_c$ and is denoted by $J$, and is denoted by $J_c$, that is:

$$J_c = \{ z \in C : \left| \left( f_c^{[n]} \right)(z) \right| > 1, \text{ and } f_c^{[n]}(z) = 0, n = 1, 2, \ldots \}.$$  

Julia sets are named for the French mathematician Gaston Julia.


The filled Julia set for $f_c(z) = z^2 + c$, where $c$ is a complex parameter is the collection of complex number $z$, whose orbit under $f_c$ is bounded, and denoted by $F_c$. That is:

$$F_c = \{ z \in C : \lim \left( f_c^{[n]}(z) \rightarrow \infty \right) \}.$$  

The reminder of the research paper is organized as follows: section 1 introduces the concepts needed to understand the correlation function for calculating the correlation dimension $D_{cor}$. In section 2, we present the Escape Time Algorithm. In section 3, we present the proposed method to compute the correlation dimension of the Filled Julia set generated by Escape Time Algorithm. Finally, the conclusions are drawn in section 4.

1. **Correlation dimension:**

There are many ways to define the fractal dimension, but one of the numerically simplest and most widely used is the correlation dimension of Grassberger and Procaccia (1983a,b). The real interest of the correlation dimension is determining the dimensions of the fractal. There are other methods of measuring dimension such that Hausdorff dimension, the box-counting dimension but the correlation dimension has the advantage of being straightforward and quickly calculated, when only a small number of points is available, and is overwhelmingly consensual with other calculations dimension.

**Definition 3:**

Let $\{x_i\}_{i=1}^{N}$ be a sequence of points in $\mathbb{R}^n$. The correlation function, $C(\varepsilon)$ is defined by:

$$C(\varepsilon) = \lim_{N \rightarrow \infty} \frac{1}{N^2} \times \{ \text{Number of pairs } (x_i, x_j) \text{ of points with } d(x_i, x_j) < \varepsilon \}$$

$$C(\varepsilon) = \lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{i,j} \theta(\varepsilon - ||x_i - x_j||),$$

where $\theta(x)$ is the Heaviside step function (unit step function) which has a value of either 0 or 1 and may be defined as:

$$\theta(\varepsilon - ||x_i - x_j||) = \begin{cases} 1, & 0 \leq \theta(\varepsilon - ||x_i - x_j||) \\ 0, & 0 > \theta(\varepsilon - ||x_i - x_j||) \end{cases}$$

which acts as a counter of the number of pairs of points with Euclidean separation - distance between two points on the fractal, $x_i$ and $x_j$. The multiplier $\frac{1}{N^2}$ is included to normalize the count by the number of pairs of points on the fractal.

Grassberger and Procaccia (1983a,b) and Barnsley (1993) established that for small values of the separation distance $\varepsilon$, the correlation function $C(\varepsilon)$ has been found to follow a power law such that

$$C(\varepsilon) \sim C \varepsilon^{-D_{cor}}$$

where $C(\varepsilon)$ is the number of cells with the edge size $\varepsilon$ necessary to cover all points of the fractal object and $C$ is a positive constant. The “$\sim$” symbol in this expression is used to indicate that this is not an exact equality but is a scaling relation that expected to be valid for sufficiently large $N$ and small $\varepsilon$. After taking logarithms of each side of the scaling relation and rearranging terms, we define

$$D_{cor} = \lim_{\varepsilon \rightarrow 0} \frac{\log(C(\varepsilon))}{\log(\varepsilon)}.$$  

The correlation dimension $D_{cor}$, estimated using the least squares linear regression of $\log C(\varepsilon)$ and versus $\log(\varepsilon)$, then the slope of linear model represent $D_{cor}$.

2. **The Escape Time Algorithm** (Barnsley, M.F., 1993; Mohammed, A.J., 2005):

In this section, we study fractals generated by means of Escape time algorithms (ETA). Such fractals are computed by repeatedly applying a transformation to a given initial point in the plane, the algorithm is based on the number of iterations necessary to determine whether the orbit sequence tends to infinity or not that is, an orbit diverges when its points grow, further apart without bounds. Escape time algorithm is numerical computer
The Algorithm

1. Given \( \mathbb{W} \subseteq \mathbb{R}^2 \), s.t \( \mathbb{W} = \{(x, y) : a \leq x \leq c, b \leq y \leq d\} \), define the array of point in \( \mathbb{W} \) by \( x_{p,q} = (a + p(cs - as)/M, b + q(ds - bs)/M), \) \( p, q = 1, 2, \ldots, M \), for any positive integer \( M \).

2. Let \( \mathbb{C} \) be a circle centered at the origin, the set \( \mathbb{R} \) is defined such that, \( \mathbb{R} = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 > r^2\} \), where \( r = 0.5\sqrt{0.25 + \sqrt{a^2 + b^2}} \).

3. Let \( f_c \) be a complex function with the orbit of point \( \{f_c^{[n]}(x_{p,q})\}_{n=0}^\infty \), where \( (x_{p,q}) \in \mathbb{W} \).

4. Repeat, \( \forall x_{p,q} \in \mathbb{W} \).

IF \( \{f_c^{[n]}(x_{p,q})\}_{n=0}^\infty \in \mathbb{R} \), then \( x_{p,q} \) is colored with the colored with white color.

Else it is colored Black.

the Filled Julia set generated by escape time point \( \mathbb{W} \) is defined as

\[ F_c = \{x_{p,q} \in \mathbb{W} : f_c^{[n]}(x_{p,q}) \notin \mathbb{R} \text{ for all } n \leq N\} \]

\( = \{x_{p,q} \in \mathbb{W} \text{ such that } x_{p,q} \text{ is black point}\} \).

Output: The set \( F_c \) is called the fractal constructed by Escape Time Algorithm.

--------Calculate correlation dimension (CorD)--------

Step1: INPUT \( N_t, N \) \( \{N_t = \text{No. of transients}, N = \text{NO. of points}\} \)

Step2: Generate random \( x_i, y_i, i = 1:N_t \)

Step3: \( d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, m_{nx} = \min(d), mx_x = \max(d) \)

Step4: \( T = \frac{2\log(mx_x)}{\log(2)}, n_x = \left[\frac{\log(T)}{\log(2)}\right] + 1 \)

Step5: \( \epsilon v_i = \frac{2(mx_x)}{\epsilon} (-i - 1), i = 1, \ldots, n_x \)

Step6: \( N_p = N^{N(N-1)/2} \)

\( N_j = [d < \epsilon v_i, d > 0], \mathcal{S} = \sum N_j, c_{e_i} = \text{concate} [c_{e_i}, S/N_p], i = 1:n_x \)

Step7: \( O_p = 3, k_1 = O_p + 1, k_2 = n_x - O_p \)

\( x_{di} = \frac{\log(c_{e_i})}{\log(2)}, y_{di} = \frac{\log(ce_i)}{\log(2)}, x_p = x_{di}, y_p = y_{di}, i = k_1:k_2 \)

Step8: \( a_4x + a_0 = \text{Polylt} \left( x_p, y_p, 1 \right), \text{CorD} = a_1 \)
32
Arkan Jassim Mohammed and Noora Ahmed Mohammed, 2016

OUTPUT: PRINT (CorD):
3. Computing Correlation Dimension of the Filled Julia Set:

In this section, we present an approach algorithm to compute correlation dimension of fractals generated by Escape Time Algorithm and the correlation function has been investigated by Grassberger–Procaccia Algorithm (1983a,b).

A log-log graph of the correlation function \( C(\varepsilon) \) versus the distances between every pair of points \( \varepsilon \) in the fractals constructed by Escape Time Algorithm is an approximate of the correlation dimension, i.e., the correlation dimension is deduced from the slope of the straight line scaling grid in a plot of \( \log C(\varepsilon) \) versus \( \log(\varepsilon) \).

We presented and implemented in the Matlab program listed in Appendix for computing the correlation dimension of fractals generated by Escape Time Algorithm.

for finding the correlation dimension for filled julia set \( F_c \) for each \( c \in \mathbb{C} \), generated by Escape Time Algorithm, which is listed in the Appendix. By applying these program for different types of \( c \).

Example1:

Let the function \( f_{c=1.1} : \mathbb{C} \to \mathbb{C} \), be given by the formula: 

\[ f_{c=1.1}(x, y) = (x^2 - y^2 - 1.1, 2xy) \]

for all \((x, y) \in \mathbb{C}\).

Define \( R \) by choosing \( r = 4 \), and let \( W = \{(x, y) : -2 \leq x \leq 2, -2 \leq y \leq 2\} \).

The result of running the Escape Time Algorithm, with \( R, W \) and \( f : \mathbb{C} \to \mathbb{C} \), thus defined is shown in figure (1.a). The black object represents the filled Julia set \( F_{c=1.1} \), and we find the correlation dimension by this figure (1.b) show the plot of \( \log C(\varepsilon) \) against \( \log(\varepsilon) \) for the fractal. Correlation dimension is computed from the sloped linear portion of the plot determined from a least square fit. \( (D_{cor}(F_{c=1.1}) = 1.5043) \).

![Fig. 1.a: The filled julia set of quadratic \( f_c = z^2 + c, \ c = 1.1 \)](image)

![Fig. 1.b: Correlation dimension for filled julia set generated by Escape Time Algorithm](image)
Example 2:
Let the function $f_{c=1.3} : \mathbb{C} \to \mathbb{C}$, be given by the formula: $f_{c=1.3}(x, y) = (x^2 - y^2 - 1.3, 2xy)$ for all $(x, y) \in \mathbb{C}$.

Define $\mathcal{R}$ by choosing $r = 4$, and let $\mathcal{W} = \{(x, y) : -2 \leq x \leq 2, -2 \leq y \leq 2\}$.

The result of running the Escape Time Algorithm, with $\mathcal{R}$, $\mathcal{W}$ and $f : \mathbb{C} \to \mathbb{C}$, thus defined is shown in figure (2.a). The black object represents the filled Julia set $F_{c=1.3}$, and we find the correlation dimension by this figure (2.b) show the plot of $\log C(\varepsilon)$ against $\log(\varepsilon)$ for the fractal. ($D_{cor}(F_{c=1.3}) = 1.5161$).

**Fig. 2.a:** The filled julia set of a quadratic $f_c = z^2 + c$, $c = 1.3$

correlation dimension for filled julia set generated by Escape Time Algoritm

Example 3:
Let the function $f_{c=1.6} : \mathbb{C} \to \mathbb{C}$, be given by the formula: $f_{c=1.6}(x, y) = (x^2 - y^2 - 1.6, 2xy)$ for all $(x, y) \in \mathbb{C}$.

Define $\mathcal{R}$ by choosing $r = 4$, and let $\mathcal{W} = \{(x, y) : -2 \leq x \leq 2, -2 \leq y \leq 2\}$.

The result of running the Escape Time Algorithm, with $\mathcal{R}$, $\mathcal{W}$ and $f : \mathbb{C} \to \mathbb{C}$, thus defined is shown in figure (3.a). The black object represents the filled Julia set $F_{c=1.6}$, and we find the correlation dimension by this figure (3.b) show the plot of $\log C(\varepsilon)$ against $\log(\varepsilon)$ for the fractal. ($D_{cor}(F_{c=1.6}) = 1.4937$).
4. Conclusion:

In this research paper we have presented an algorithm for computing correlation dimension $D_{cor}$ of Filled Julia set ($F_c$) generated by Escape Time Algorithm (ETA), which is based on computing the correlation function. Computing the correlation function based on the selection of the Euclidean distance is presented and implemented the Matlab program listed in the Appendix for reducing the computational complexity of the Grassberger-procaccia Algorithm.

Appendix:

--- Computer program for computing the correlation dimension of the Filled Julia set generated by Escape Time Algorithm

```matlab
clc; clear all;
as=2.5; bs=-2.5; cs=1.5; ds=-1.5;
c=[1.6 0];
a=[1 0 1 0 0 -1 0 0 0 2 0 0];
a(1)=c(1); a(7)=c(2);
umits = 5; R = 1; mm=140;
m=28;
A1=0; t=0; h=0;
```

---
for p = 1:mm
for q = 1:mm
x = a + (b-a)*p/m;
y = b + (d-b)*q/m;
for n = 1:numits;
% u=x^2-y^2-a;
% v=2*x*y-b;
newx=a(1)+a(2)*x+a(3)*x.^2+a(4)*x*y+a(5)*y+a(6)*y.^2;
newy=a(7)+a(8)*x+a(9)*x.^2+a(10)*x*y+a(11)*y+a(12)*y.^2;
x=newx;
y=newy;
if u^2 + v^2 <= r^2
if abs(x)^2+abs(y)^2 > 100
%if abs(x)^2 > 100
 t=t+1;
P(t)=p; Q(t)=q;
end;
else
 h=h+1;
P1(h)=p; Q1(h)=q;
end;
end;
f(x^2 + y^2)^0.5 > R^2
A1 = A1 + 1;
end;
end;
da=-2; b=2;
for j=1:100
% x(j) = a + (b-a).*rand(1,1); y(j) = a + (b-a).*rand(1,1);
Me=(abs(X(j))+abs(Y(j)))/2;
%fprintf('%4.3f, %4.3f',X(j),Y(j),Me);
x(j)=X(j)/Me; y(j)=Y(j)/Me;
fprintf('%4.3f , %4.3f',x(j),y(j));
end;
D1 = log(A1)/n*log(2)
figure(1)
plot(P(1:t),Q(1:t),'.black','MarkerSize',5);
xlabel('itx_n')
ylabel('ity_n')
title(' ETA for filled Julia Set');
figure(2)
plot(X(1:t),Y(1:t),'.black','MarkerSize',5);
xlabel('itx_n')
ylabel('ity_n')
title(' ETA for filled Julia Set');
N_pts=100;
ED=sparse(N_pts,N_pts);
k=0;
for j=1:100 %N_pts
%fprintf('%4.3f , %4.3f\n',x(j),y(j));
for i=j+1:100 %N_pts
d=(x(i)-x(j))^2+(y(i)-y(j))^2;
ED(i,j)=d;
k=k+1;
if mod(k,250000)==0
%fprintf('%d - %3.2f\n',k,d);
end;
end; end;
ED=sqrt(ED);
min_eps=double(min(min(ED+(1000*ED==0))));
m_eps=double(max(max(ED)));
max_eps=2^ceil(log(m_eps)/log(2));
n_div=floor(double(log(max_eps/min_eps)/log(2)));
n_eps=n_div+1;
eps_vec=max_eps*2.^(-(1:n_eps)-1));
Npairs=N_pts*(N_pts-1)/2;
c_eps=[];
for i=1:n_eps
eps=eps_vec(i);
N=(ED<eps & ED>0);
S =double(sum(sum(N)));
c_eps = [c_eps; S/Npairs];
end;
omit_pts=3;
k1=omit_pts+1; k2=n_eps-omit_pts;
in_grid = k1:k2;
xd=log(eps_vec)/log(2);
yd=log(c_eps)/log(2);
[xp,yp]=xd(in_grid); yp=yd(in_grid);
[coeff,temp]=polyfit(xp,yp,1);
D_C=coeff(1);
poly=D_C*xd+coeff(2);
figure(3);
plot(xd,yd,'s-','MarkerSize',5);
hold on
plot(xd,poly,'r-','MarkerSize',5);
axis tight
plot([xd(k1),xd(k1)],[20,0],'m--');
plot([xd(k2),xd(k2)],[20,0],'k--');
xlabel('log_2(\epsilon)');
ylabel('log_2(C(\epsilon))');
title('correlation dimention for filled julia set generated by Escape Time Algorithm');
grid

REFERENCES