Fractals Generating Techniques

Ahammad Hossain[a], Nurujjaman [1,*], Rita Akter[2], Dr. Payer Ahmed[3]
[1]Department of Mathematics, Sonargaon University, Dhaka,
*Corresponding Author: Department of Mathematics, Sonargaon University, Dhaka.
E-mail:md.jaman24@gmail.com.com, md.jaman@su.edu.bd

Abstract

Fractals Generating Techniques introduce interesting part of Fractals Geometry. In this paper, we introduce some outstanding beautiful images known as Fractals. Our goal is to show techniques to generate some beautiful fractals like Mandelbrot Set, Fractal Trees, Heart Shape Fractal, Julia set, Height Field. We restrict our attention to generate the said spectacularly images considering some techniques such as geometric iteration rules, successive removals etc. Special emphasize is given to consider very new generating functions as well as their suitable initial seeds so that we can see some new fractals after a number of iterations. Necessary programs are considered for all cases. We use Mathematica and Mat Lab to perform programming.

Keywords: Fractal, Iteration, Mandelbrot Set, Heart Shape, Julia set, Height Field.

1.0 Introduction

The term 'Fractal' is fascinating to many people which are none other than the beautiful but complicated images in the nature. Typically, mathematics consists of complicated figures, boring formulas and often monotonous calculation while the fractal geometry brings art in the field of mathematics which gives a different taste of the study. The most interesting thing about fractal is that they give a mathematical description of the existing natural object which often includes very complicated patterns such as coastlines, mountains, ferns, trees or parts of living organisms [1]. Before the invention of computer some people had done a tremendous work on fractals though fractal geometry is closely connected with computer techniques. At first the British cartographers encountered the problem in measuring the length of Britain coast. The actual length of the coastline was approximately half the length of coastline measured on a detailed map [2]. As they looked closer and closer they found more detailed and longer the coastline. Without realizing they had discovered one of the main properties of fractals.

1.1 Historical background

The credit goes to Benoît Mandelbrot for the development of fractal geometry; many other mathematicians preceding him in the century had laid the foundations for his work. Moreover, Mandelbrot was able to utilize the advancements of computer technology that his predecessors distinctly lacked; however, this in no way diminishes from his visionary achievements. Nevertheless, to make Mandelbrot's work clearer and to establish its connections to other branches of mathematics the works of Karl Weierstrass, Georg Cantor, Felix Hausdorff, Gaston Julia, Pierre Fatou and Paul Lévy undoubtedly helps in a salient way [3].

1.2 Basic idea

The idea of fractals is comparatively new, but the seed was sowed in the 19th century mathematics [3]. A fractal is a fragmented shape that can be subdivides in part each of which at least a reduced size copy of the whole. Mathematically we can generate fractals which are reproducible at any magnification or reduction and the reproduction of each parts looks just like the original, or at least has a similar pattern.

The familiar Euclidean geometry deals with objects which includes integer dimensions such as zero-dimensional points, one-dimensional lines and curves, two-dimensional surfaces like planes, and three-dimensional solid objects such as balls and blocks (e.g. spheres and cubes). However, the objects found in the nature which often have dimensions are not a whole number. And the reason for this is the property called self-similarity.

A fractal is an ongoing pattern. Fractal patterns are infinitely complicated and they are self-similar across different scales. It includes a very simple method to create a fractal. If repeat a simple pattern over and over again continuously we come to end with a fractal. This is an ongoing feedback loop. Driven by recursion, fractals are images of dynamic systems – the pictures of Chaos [4]. Geometrically, they lie in between our familiar dimensions. Fractal patterns are very mush familiar to us as the nature is full of fractals. For example: clouds, rivers, trees, mountains, coastlines, seashells, hurricanes, etc. While the abstract fractals such as the Mandelbrot Set which can be generated by repeating a simple complex function repeatedly.

In this paper, we will describe some of the wonderful new ideas in the area of mathematics known as fractal geometry. As we will see, fractals are incredibly complicated and often quite beautiful geometric shapes that can be generated by simple rules.

The word is related to the Latin verb frangere, which means "to break" [4]. In the Raman mind, frangere may have evoked the action of breaking a stone; since the adjective derived it combines the two most obvious properties of broken stones, irregular and fragmentation. This adjective is fractus, which lead to fractal. The etymological kinship with "fraction" is also significant if ones interprets "fraction" as a number that lies between integers. Indeed, a fractal set can be considered as lying between the shapes of Euclid [5].

In his founding paper Beno^it Mandelbrot coined the term Fractal, and described it as follows:

A [fractal is a] rough or fragmented geometric shape that can be subdivided in parts, each of which is (at least approximately) a reduced-size copy of the whole [6].

1.3 Some famous fractals

There are a lot of fractals that have generated by the mathematicians. Among them some most famous fractals are the Sierpinski Triangle, the Koch Curve, and the Cantor Set etc.

Here we discussed another two famous fractals—The Mandelbrot and Julia set.

1.3.1 The man delbrot set

Named after Benoit Mandelbrot, Among the existing fractals the Mandelbrot set is one of the most famous and complicated fractal. Behind this complex picture there is a simple equation. Mandelbrot was playing with the simple quadratic equation $z=z^2+c$ and made the most famous fractal in the history. Both z and c are complex numbers in this equation. In other words, the Mandelbrot set is the set of all complex c such that iterating $z=z^2+c$ does not diverge [6].

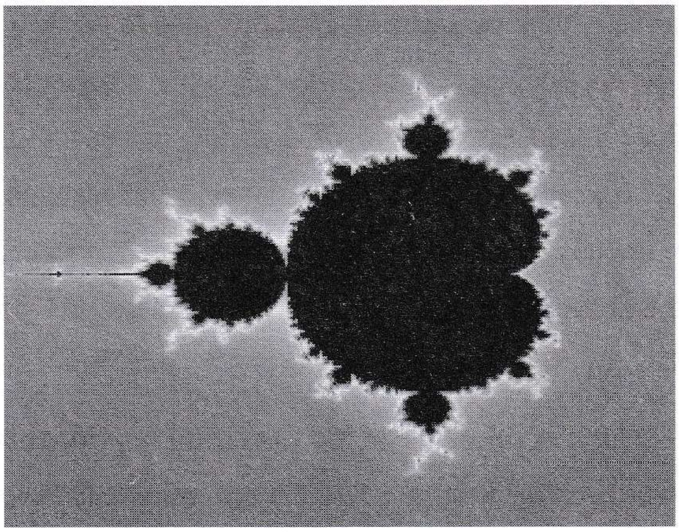

Fig. 1. Mandelbrot Set

To generate the Mandelbrot set graphically, the computer screen is to be considered as the complex plane. Each point on the plane have to tested by the equation $z=z^2+c$. If the iterated z stayed within a given boundary forever that is it converges then the point is inside the set and the point is plotted black. If the iteration went of control that is it diverges then the point was plotted in a color with respect to how quickly it vanishes.

For instance, let c = 1 gives the sequence 0, 1, 2, 5, 26... and clearly it tends to infinity. Since the sequence is unbounded for 1, so 1 is not an element of the Mandelbrot set. On the other hand if c = i (where i is defined as $i^2 = -1$) gives the sequence 0, i, (-1 + i), -i, (-1 + i), -i, ... which is bounded and so i belongs to the Mandelbrot set.

1.3.2 The Julia set

Another famous fractal which is very closely related to the Mandelbrot set is the Julia set. It was named after Gaston Julia [6,7], during the early twentieth century who studied the iteration of polynomials and rational functions, making the Julia set much older than the Mandelbrot set

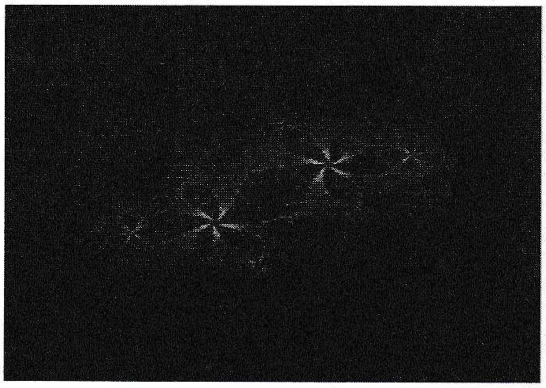

Fig. 2. Julia Set

The remarkable difference between the Julia set and the Mandelbrot set is in the way of iteration. In the case of Mandelbrot set we have to iterate z always starting from 0 and varying the value of c. Where the Julia set iterates for a fixed value of c and varying values of z. That is we can say that, the Mandelbrot set is in the parameter space, or the c-plane, while the Julia set is in the dynamical space, or the z-plane [6,7].

2.0 Methodology

Where Euclidean geometry describes lines, ellipses, circles, etc. with equations, fractal geometry describes objects in terms of algorithms that are sets of instructions on how to create a fractal. One way to describe fractals is through what are called iterated function systems, or IFS [8]. This is the only type of fractal that we shall discuss in detail in this thesis paper. IFS follow the general approach of altering a geometric object in a particular way, leaving multiple smaller objects each of which is similar to the original, and then repeating the process on each of those smaller objects to create even smaller parts, and so on. The fractal is the result of carrying this process out infinitely many times.

* The iterated function systems is used based on fixed geometric replacement rules; may be stochastic or deterministic; e.g., Koch snowflake, Cantor set, Haferman carpet, Sierpinski carpet, Sierpinski gasket, Peano curve, Harter-Heighway dragon curve, T-Square, Menger sponge

* Strange attractor is the method which includes iterations of a map or solutions of a system of initial-value differential equations that exhibit chaos.

* Escape-time fractals is a formula or recurrence relation at each point in a space (such as the complex plane); usually quasi-self-similar; also known as "orbit" fractals; e.g., the Mandelbrot set, Julia set, Burning Ship fractal, Nova fractal and Lyapunov fractal. The two-dimensional vector fields that are generated by one or two iterations of escape-time formulae also give rise to a fractal form when points (or pixel data) are passed through this field repeatedly.

* Stochastic rules generate random fractals; such as the Lévy flight, percolation clusters, self-avoiding walks, fractal landscapes, trajectories of Brownian motion and the Brownian tree (i.e. dendritic fractals can be generating by modeling diffusion-limited aggregation or reaction-limited aggregation clusters).

* A recursive topological algorithm for refining tiling includes finite subdivision rules and this is same as the process of cell division. For instance, the Cantor set and the Sierpinski carpet are generated by iterative processes which includes finite subdivision rules, as is barycentric subdivision [9].

There are some of senses to show techniques to generate some beautiful fractals like Mandelbrot Set, Julia set, Pythagorean Tree, Heart Shape Fractal, fractal Crown, Height Field.
We use some mathematical software's like MATHEMATICA, MATLAB etc. so that we can describe the graphical representation of our mathematical research.

3.0 Experiments & results

Considering different and suitable functions, we have eventually generated some beautiful and natural images. These images include beautiful flowers, household things, ornaments fractal and fractals that result from known and famous mathematical functions or combination of them. On the basis of MATHEMATICA and MATLAB Program we present here some beautiful and interesting fractals.

3.1 Pythagorean Tree

Several steps for construction of Pythagorean tree are shown below.

We replace each 'branch' of 1st step with a scaled copy of the generator to create 2[nd] iteration.

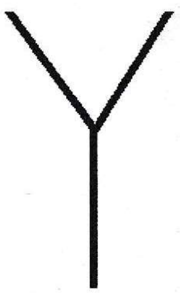

Fig. 3. Step 1

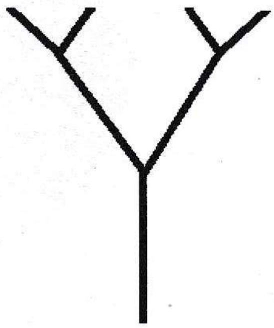

Fig. 4. Step 2

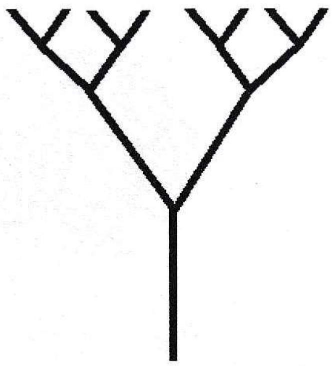

Fig. 5. Step 3

We can repeat this process to create later steps.

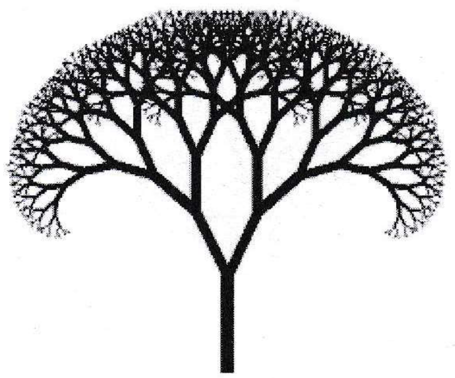

Fig. 6. Final step.

Repeating this process, we can create Pythagorean tree.

3.1.1 MATHEMATICA Program

```
FractalTree[pt : {_, _}, \[Theta]orient_: \[Pi]/2, \[Theta]sep_: \[Pi]/9,
        depth_Integer: 9] := Module[{pt2},
If[depth == 0, Return[]];
        pt2 = pt + {Cos[\[Theta]orient], Sin[\[Theta]orient]}*depth;
        DeleteCases[ Flatten@{Line[{pt, pt2}],
                fractalTree[pt2, \[Theta]orient - \[Theta]sep, \[Theta]sep,depth - 1],
                fractalTree[pt2, \[Theta]orient + \[Theta]sep, \[Theta]sep,depth -1]},
                Null]]
Graphics [fractalTree[{0, 0}, \[Pi]/2, \[Pi]/9]]
```

3.2 Heart Shape Fractal

Generating Function: $320((-x^2z^3 - \frac{9y^2z^3}{80}) + (x^2 + \frac{9y^2}{4} + z^2 - 1)^3) = 0$

The function forms the heart shape in 3D with appropriate MATLAB Code that has been given below.

At 3rd iteration it takes the form of a rectangle rotating about 45 degrees. At 10th iteration it becomes to look like octagonal.

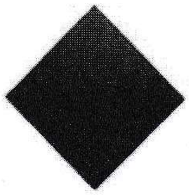

Fig. 7. At 3rd Iteration

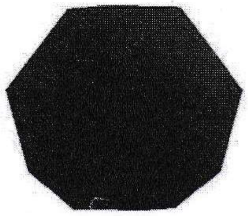

Fig. 8. At 10th Iteration

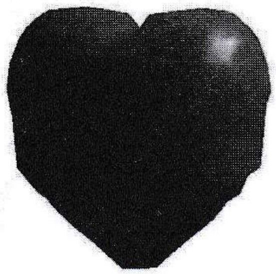

Fig. 9. At 30thIteration

When the iteration number gets the score 100 the function finally produces actual heart shape image.

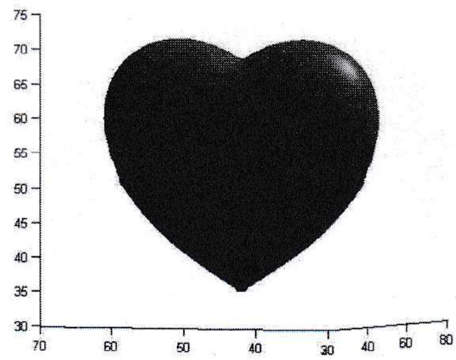

Fig. 10. After 100th iteration

3.2.1 MATLAB Program

```
% set up mesh
n=100;
x=linspace(-3,3,n);
y=linspace(-3,3,n);
z=linspace(-3,3,n);
[X,Y,Z]=ndgrid(x,y,z);
%Compute function at every point in mesh
F=320 * ((-X.^2 .* Z.^3 -9.*Y.^2.*Z.^3/80) + (X.^2 + 9.* Y.^2/4 + Z.^2-1).^3);
%generate plot
isosurface(F,0)
```

view ([-67.5 2]);
colormap(flag);

3.3 Fractal Crown

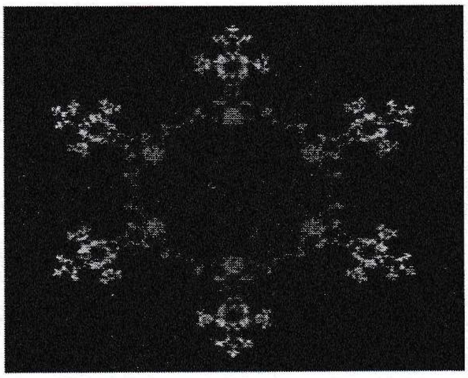

Fig. 11. Fractal crown

Generating Function: $\sum_{k=1}^{14} \frac{e^{i(-a)^k} t}{e^{bk}}; \quad a = 0.5; b = \frac{\log 2}{\log 3}$

3.3.1 MATHEMATICA Program

```
n = 280;
a = 5.0;
b = Log[2]/Log[3];
image = Table[0, {n}, {n}];
Do[w = Sum[E^(I (-a)^k t)/a^(b k), {k, 1, 14}];
{i, j} = Floor[n({Re[w], Im[w]}/1.25 + 0.5)];
image[[i, j]] = Abs[w], {t, -Pi, Pi, 0.001}];;
ListDensityPlot[image, Mesh -> False, Frame -> False]
```

3.4 Generating Mandelbrot set

Fig. 12. G.F. f(z)=z+c, c=-.2+0i

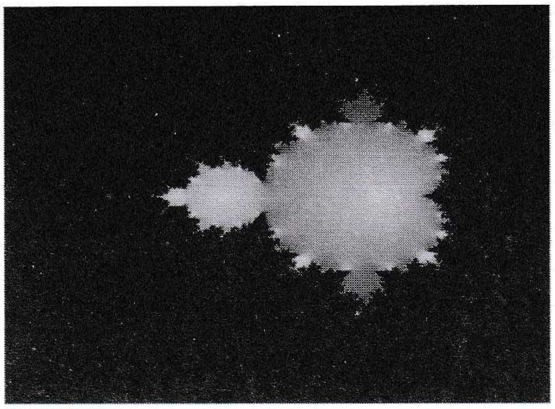

Fig. 13. G. F. $f(z)=z^2+c$, c=-.6+0i

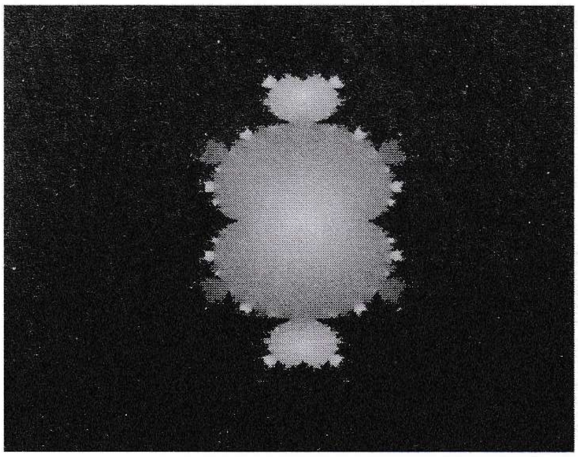

Fig. 14. G.F. $f(z)= z^3 +c$, c=.6+0i

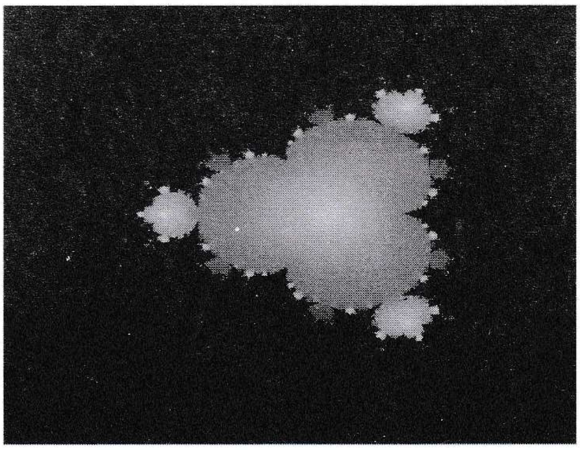

Fig. 15. G. F. $f(z)=z^4+c$, c=-.6+0i

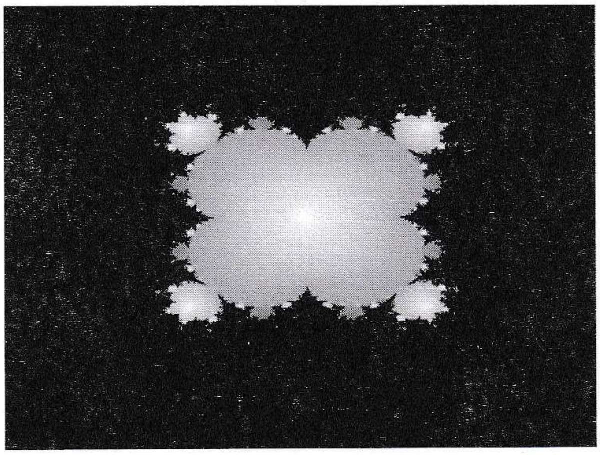

Fig. 16. G. F. $f(z)=z^5+c$, $c=-.6+0i$

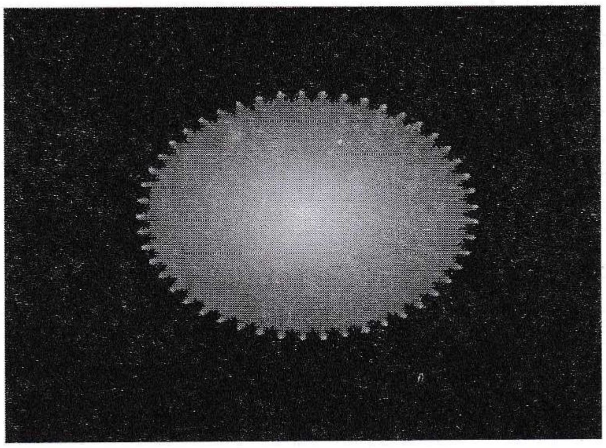

Fig. 17. G. F: $f(z)=z^{50}+c$, $c = -.2+0i$

3.4.1 MATLAB Program:

```
col=20;
m=400;
cx=-.2;
cy=0;
l=1.5;
x=linspace(cx-l,cx+l,m);
y=linspace(cy-l,cy+l,m);
[X,Y]=meshgrid(x,y);
Z=zeros(m);
C=X+i*Y;
for k=1:col;
Z=Z.^50+C;
W=exp(-abs(Z));
end
colormap copper (256);
pcolor(W);
shading flat;
axis('square','equal','off');
```

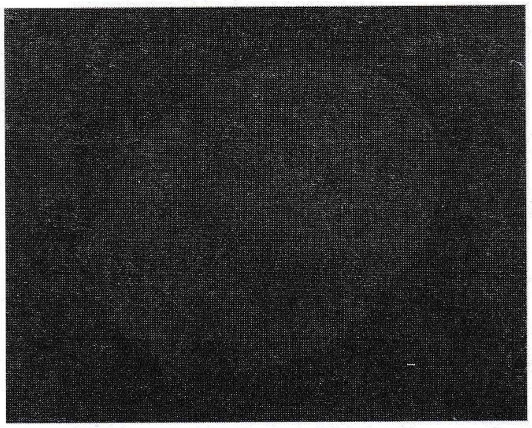

Fig. 18. 1st iteration (when $z = 0$)

3.5 Generating Julia set

Generating Function:

$f(z) = z^2+ c$ where $c = 0.27+0.53i$.

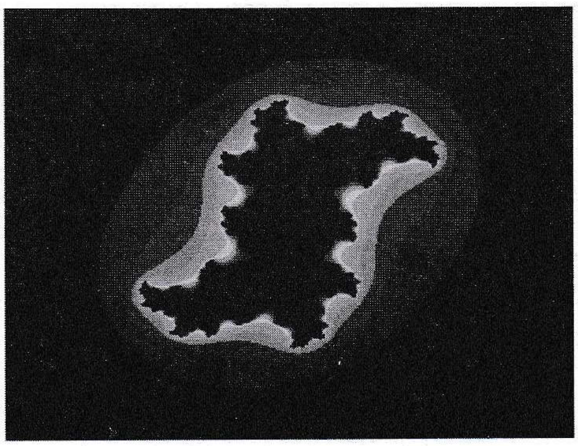

Fig. 19. 10th iteration

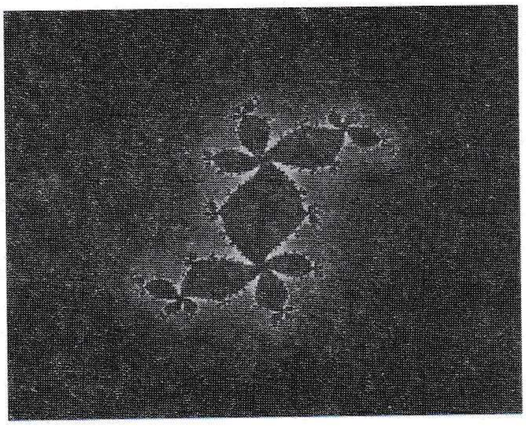

Fig. 20. 50th iteration

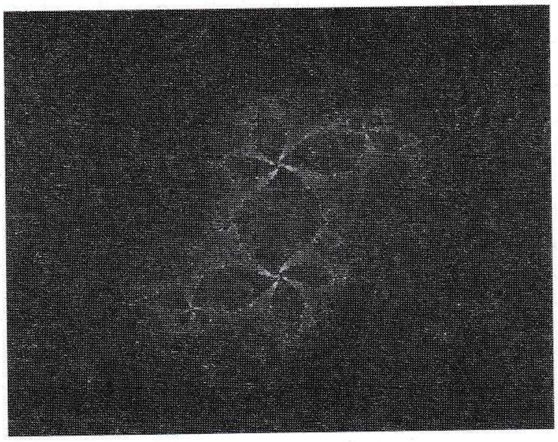

Fig. 21. 120th iteration: Complete Julia set.

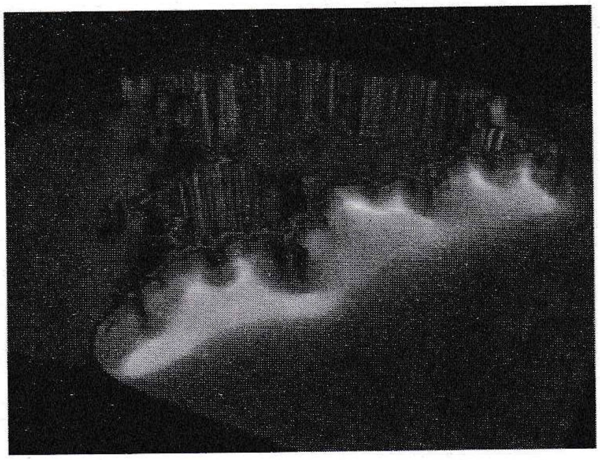

Fig. 22. Mandelbrot set height field.

3.5.1 MATLAB Program:

```
%%% Compute and draw the Julia set
clear;
clc;
%%% Parameters
c = 0.27+0.53i;          % complex number
niter=1;                 % number of iterations
th=10;                   % threshold to determine divergence
v=1000;                  % resolution (<-> number of points to compute)
%%% Initialisation
r = max(abs(c),2);    % radius of the circle beyond which every point diverges
d = linspace(-r,r,v);  % divide the x-axis
Z = ones(v,1)*d+i*(ones(v,1)*d)';   % create the matrix A containing complex numbers
C = zeros(v,v);  % Julia set point matrix
%%% Compute the julia set
```

```
for k = 1:niter
Z = Z.*Z+ones(v,v).*c;
C = C+(abs(Z)<=r);
end
%%% Figurefigure(21)
clf;
imagesc(C);
colormap(jet);
hold off;
axis equal;
axis off;
```

3.6 Generating Mandelbrot Set Height Field:

Generating Function: $cet = n + \log_2 \ln(R) - \log_2 \ln|z|$

3.6.1 MATHEMATICA Program

```
R = 6;
image = ParametricPlot3D[Module[{z = 0.0, i = 0}, While[i < 100 && Abs[z] < R^2, z = z^2 + xc
+ I yc; i++]; cet = If[i != 100, i + (Log[Log[R]] - Log[Log[Abs[z]]])/Log[2], 0]; {xc, yc,
0.5Min[0.1cet, 1], {EdgeForm[], SurfaceColor[Hue[1 - 0.1cet]]}}], {xc, -2.0, 1.0}, {yc, -1.5, 1.5},
PlotPoints -> 64, Boxed -> False, Axes -> False, DisplayFunction -> Identity];
<< MathGL3d`OpenGLViewer`;
MVShow3D[image, MVNewScene -> True];
```

4.0 Conclusion

Fractals Generating Techniques introduce interesting part of Fractals Geometry. In this thesis, we introduce some outstanding beautiful images known as Fractals. Our goal is to show techniques to generate some beautiful fractals like Mandelbrot Set Fractal, Fractal Trees, Julia set, Pythagorean Tree, Heart Shape Fractal, fractal Crown, Height Field. We restrict our attention to generate the said spectacularly images considering some techniques such as geometric iteration rules, successive removals etc. Special emphasize is given to consider very new generating functions as well as their suitable initial seeds so that we can see some new fractals after a number of iterations. Necessary programs are considered for all cases. We still are failing to consider generating function or suitable initial seed for some fractals though their images exist.

References

[1] Patrzalek, E., 2004, General Introduction to Fractal Geometry, Stan Ackermans Institute, IPO, Centre for User-System Interaction, Eindhoven University of Technology, Chap. 1.

[2] Crownover, R. M., c1995, Boston: Jones and Bartlett, Introduction to Fractals and Chaos, Chap. 3.

[3] Trochet, H., A History of Fractal Geometry, University of St Andrews, Chap. 1.

[4] Devaney, R. L., 1986, MenloPark: Benjamin/Cummings, An Introduction to Chaotic Dynamical Systems, Chap. 4.

[5] Lefler, S., 7-1-2006, University of Nebraska-Lincoln, Fractals and the Chaos Game, Chap .2.

[6] Mandelbrot, B. B., 1989, Printed in Great Britain, "Fractal geometry: what is it, and what does it do?", p. 16.

[7] Rostami, R., Amir, M., Azar, 1988, Fractals and Fractal Geometry, Chap. 1-3.

[8] Shrivastava, S. C., 2016, International Advanced Research Journal in Science, Engineering and Technology, "Iterated Function System", vol-3, issue-8, p. 263-255.

[9] Khanbareh, H., 22 December 2011, "Fractal Dimension Analysis of Grain Boundaries of 7XXX Aluminum Alloys and Its Relationship to Fracture Toughness" p. 77.