

Enhancing Computational Thinking with Spreadsheet and Fractal Geometry: Part 3 Mandelbrot and Julia Set

K.P Soman, Manu Unni V.G, Praveen Krishnan, V. Sowmya
Centre for excellence in Computational Engineering and Networking (CEN),
Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India

ABSTRACT

The Mandelbrot Set is the most complex object in mathematics; its admirers like to say. An eternity would not be enough time to see it all, its disks studded with prickly thorns, its spirals and filaments curling outward and around, bearing bulbous molecules that hang, infinitely variegated, like grapes on God's personal vine [1]. In this article we show how it is drawn in spread sheet. The methodology employed is same as the one used for Newton's fractal. Since it is the daddy of all fractals, a separate article is devoted to it. The same principle is extended to draw fractals based on transcendental functions.

Keywords- Computational Thinking, Mandelbrot Set, Julia Set, Fractal geometry

1. INTRODUCTION

Benoit Mandelbrot (1924-2010), a Polish-born French mathematical physicist, who was working with IBM, developed the field of **fractal geometry** which played a key role in the emergence of chaos theory. He did most of his pioneering work in the 1970s and published his findings in an illustrated and erudite book called *Fractals: Forms, Chance and Dimensions*. No one understood what he was on about - largely because the prose ways difficult to fathom. In 1977, a much-refined version of that book was republished as *The Fractal Geometry of Nature* - and fractal geometry caught the imagination of scientists.

Ever since Mandelbrot popularized fractals, students of mathematics have been intrigued by the theory behind them. Likewise, students of computer science, and especially computer graphics, have been engrossed in the beauty of fractal images. The Mandelbrot Set is a **set of complex numbers** that exhibit an interesting behaviour when run through a simple recursive formula $z_{n+1} = z_n^2 + c$ where $z_0 = c$. When this formula is applied repeatedly (with the result from one calculation used as the input to the next calculation), numbers within the Mandelbrot set will, converge towards $0+0i$ (the origin of the complex plane) or oscillate around $0+0i$.

Numbers not in the set will, after some number of iterations, diverge away from origin. The complex numbers that "escape" or "diverge" are not in the Set, but they are the interesting values when making images. The basic technique is to use the number of iterations required to make the modulus of number escape (exceeds a certain value, usually 2) as an index into a colour table and to colour the position C (complex number representing a point in complex plane) based on colour index and colour those points C which are inside the set as plain black. Fig: 1 shows the picture of Mandelbrot.

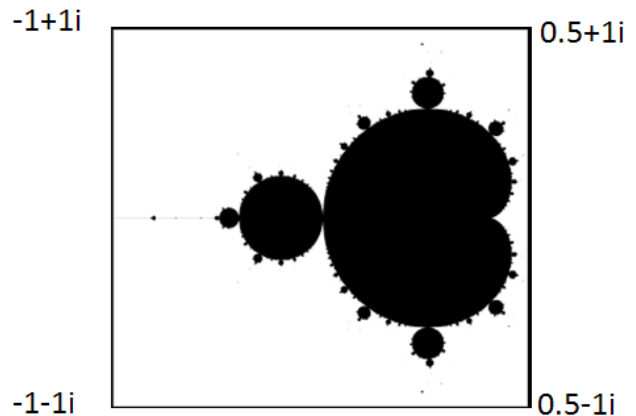


Fig:1 Mandelbrot set in complex plane

2. MATHEMATICAL DEFINITION OF MANDELBROT SET

The Mandelbrot set is a set of complex numbers defined in the following way:

$$M = \left\{ c \in \mathbb{C} \mid \lim_{n \rightarrow \infty} z_n \neq \infty \right\}$$

Where \mathbb{C} is the set of complex numbers.

$$z_0 = c ; \quad z_{n+1} = z_n^2 + c$$

That is, the Mandelbrot set is the set of all complex numbers which fulfil the condition described above, that is, if the value of the (recursive) function z_n for the value c is not infinite when n approaches infinity, then c belongs to the set.

A. How to plot Mandelbrot set?

The algorithm for defining the set is really quite simple.

1. Let c be a complex constant representing a position on screen (real part x and imaginary part y as a pair (x,y) represent a point on the screen).
2. Let $z_0 = c_x + ic_y$ be starting value of iteration.
3. Iterate once by letting $z_{n+1} = z_n^2 + c$
4. Repeat step 3 until one of the following occurs:

- The modulus (length) of z_n exceeds a given value (here we take 2 since if once modulus of z_n exceeds 2, in subsequent iteration, modulus become larger and larger). This is our test for divergence.
 - The number of iterations exceeds a given value (usually 128). This is how we assume convergence.
5. Draw the point with color determined by the number of iterations. Often one uses the following equation :Color index = $(n \bmod 16) + 1$ where n is the number of iteration taken to diverge and “ mod” represent modulo operation.

The above equation assumes the use of 16 colors with color index ‘0’ meant for the color for coloring the points which fails to diverge for the chosen iteration limit. It must be noted that choosing the proper color palette and the decision on how to choose color index for a given count of iterations to diverge (say n) is paramount importance in getting exotic looking fractals. As said earlier, there are different ways to choose colors and depending on that, figure appears differently. The boundary of the Mandelbrot set hides astonishing details. If the computation-area chosen is near boundary, the resulting figure shows fantastic patterns like the one shown below (with careful choice of the pallet).

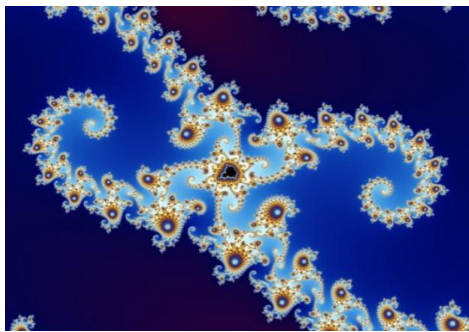


Fig.2 Mandelbrot plots

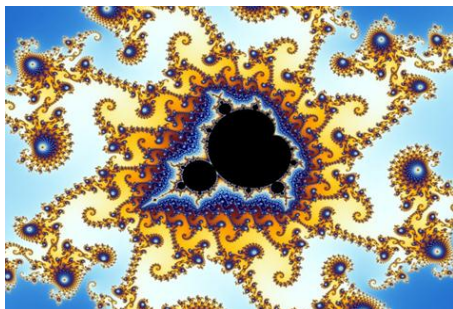


Fig.3 Mandelbrot plots

B. Good, Bad and Ugly

One main problem in web pages showing zooms of the Mandelbrot set is that very little or no attention is given to the color palette used in the images. See the one given in Fig.4. The true beauty of the Mandelbrot set is lost with poor color palette choices and lack of enough exploration.

The XaoS fractal drawing program [1] is excellent software which can zoom into the Mandelbrot set in real-time and can

produce stunningly beautiful color pictures. See plots shown in Figure 5. XaoS is free software, licensed under the GPL.

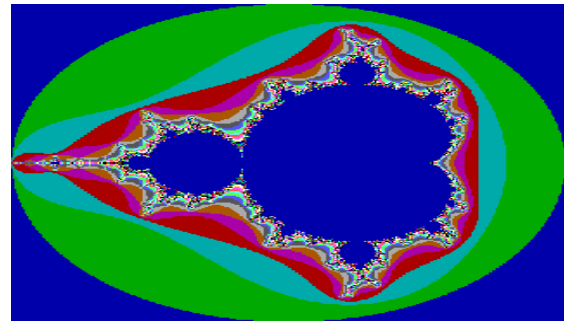
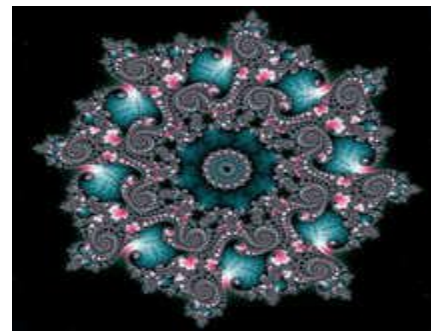
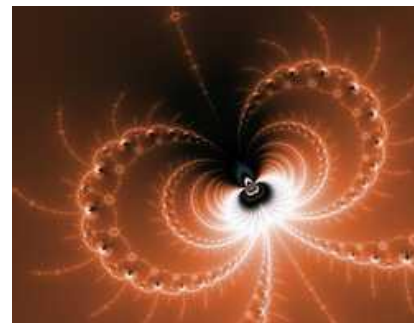


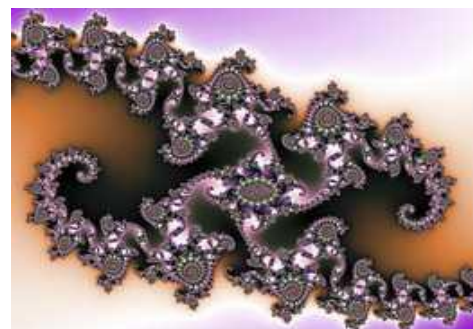
Fig.4 An Ugly plot of Mandelbrot set



(a)



(b)



(c)

Fig.5 Part of Mandelbrot Set drawn with carefully chosen color palette.

2.1 Strategy used to draw mandelbrot fractal in excel

In Microsoft Excel though we can't produce stunningly beautiful pictures, the principle behind the fractal generation can be easily explored without doing any programming. To do this, one can form a grid as shown in figure 6.. Note that each cell in the grid now gets a coordinate (x,y). It is assumed as a complex plane. So each cell's address is a complex number x+iy. So the address of the darkened cell in the grid is -0.2-0.4i.

This act as c in the iteration $z_{n+1} = z_n^2 + c$. For each cell one can do an iteration to decide its colour.



Fig.6 Excel acting as a point in Complex plane

Once the color for each cell is obtained, the cell size is reduced to that of a pixel so that the pattern can be seen as a high resolution image. The question now is how to do the iterative computation for each cell. Again Excel's What-if Utility helps us. One example iteration to excel is shown for deciding the color and then handover What-If analysis utility to take care of remaining computation.

A. Computation Involved

Let $c = c_x + ic_y$. Let the starting value for all cases be

$$z_0 = 0 + i0$$

$$z_{n+1} = z_n^2 + c \Rightarrow x_{n+1} + iy_{n+1} = (x_n + iy_n)^2 + c_x + ic_y$$

$$\text{Or } x_{n+1} + iy_{n+1} = (c_x + x_n^2 - y_n^2) + i(c_y + 2x_n y_n)$$

Therefore ,

$$x_{n+1} = c_x + x_n^2 - y_n^2 \quad (1)$$

and

$$y_{n+1} = c_y + 2x_n y_n \quad (2)$$

The iteration is carried for 32 times (for high precision we may use higher number of steps) and check the square of the magnitude of the complex number $x_{32} + iy_{32}$. If it is less than 4,

the cell corresponding to $c = c_x + ic_y$ is coloured as black.

Otherwise, colour it according to number of iteration taken to diverge. Why the limit is put 4? If any iteration, magnitude of complex number exceeds 2, it very likely that the value will diverge in succeeding iterations.

Experiment No: 1 Draw Mandelbrot fractal with two colours.

Step 1: enter Title strings

In cells A1 to A8 enter the string "n", "Cx", "Cy", "xn", "yn", "Diverged?", "Count", "Colour"

Step 2: Generate index of Iteration in a row

In B1 and C1 enter 0 and 1. Select them and drag till AH1. The values in this cell represent iteration number.

Step 3: Enter real and imaginary part of complex number "c".

In B2 and B3 enter 0.1. These values represent the real and imaginary parts of parts of complex constant "c". (You may format this cell to give some colour as it represents variable cells of What-If analysis.

Step 4: Enter starting values of iteration. (For Mandelbrot set it is c_x and c_y). Enter =B2 in the cell B4 and =B3 in cell B5. These values represent real and imaginary part of starting point of iteration.

Step 5: Enter Iteration Formula (for updating x-value).

Enter =B\$2+(B4-B5)^2 in cell C4. This is according to the formula $x_{n+1} = c_x + x_n^2 - y_n^2$. Note the dollar sign for B2. This cell value represents C_x . It should be constant throughout the iteration.

Step 6: Enter Iteration Formula (for updating y-value).

Enter =B\$3+2*B4*B5 in cell C5. This is according to the formula. $y_{n+1} = c_y + 2x_n y_n$. Note the dollar sign for B3.

This cell value represents. It should be constant throughout the iteration.

Step 7: step for 32 Iterations

Select cells C4 and C5 and drag the fill handle till AH5.

Step 8: In each iteration, check whether $\|z_n\|^2$ is diverged or not. Enter =if(C4^2+C5^2>4, "YES", "NO") in C6 and Drag till AH6.

Step 9: Count number of "No's" in row 6 (in the output of step 8) In cell B7 enter =countif(C6:AH6,"NO")

Step 10: Decide the colour of cell representing thepoint $c_x + ic_y$

. If the count in step 9 is 32 it means that the iteration did not diverge. In that case, we give colour value as 0. Otherwise, give colour value as 1. Enter =if(B7=32,0,1) in cell B8 . Now one iteration is completed.

Step 11: Prepare Table for What-if Analysis.

Most of Mandelbrot set is in the range -2 to 1 along both x and y axis

In Cell B8 and C8 enter -2 and -1.95 respectively. Select these two cell and drag till BJ11

In Cell A12 and A13, enter -2 and -1.95 respectively. Select these two cell and drag till A72

In Cell A11 (the corner cell of the table) enter =B8 (this corresponds to colour index for the example iteration).

Step 12: Use of What-if Analysis for computing colour index for different combination of x-y coordinates given by the Data-Table in step 11.

Select Cells A11:BJ72 and go for what-If Analysis. Choose Row-input cell as B2 and Column-input cell as B3. The output (partial) look as follows in Fig.7.

Step 13: Conditional formatting for colouring the cell. Select the cells in tables starting from B12 and ending in BJ72 and go for conditional formatting (available at the 'home' Tab in Excel-2010) and choose any one of the three colour palette. A part of the excel sheet look like as shown in Fig.8

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	n	0	1	2	3	4	5	6	7	8	9	10	11
2	Cx	0.1											
3	Cy	0.1											
4	xn	0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5	yn	0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
6	Diverged?	No	No	No	No	No	No	No	No	No	No	No	No
7	Count	32											
8	Color	0											
9													
10													
11	0.00	-2.00	-1.95	-1.90	-1.85	-1.80	-1.75	-1.70	-1.65	-1.60	-1.55	-1.50	-1.45
12	-2	1	1	1	1	1	1	1	1	1	1	1	1
13	-1.95	1	1	1	1	1	1	1	1	1	1	1	1
14	-1.9	1	1	1	1	1	1	1	1	1	1	1	1
15	-1.85	1	1	1	1	1	1	1	1	1	1	1	1
16	-1.8	1	1	1	1	1	1	1	1	1	1	1	1
17	-1.75	1	1	1	1	1	1	1	1	1	1	1	1

Fig.7 Snapshot of What-If Analysis for Mandelbrot fractal

	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
40	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
41	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
42	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
43	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
44	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
45	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
46	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0
47	1	1	1	1	1	1	1	0	1	0	0	0	1	1	1	1	0	0
48	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0
49	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0
50	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
51	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
53	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
55	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0
56	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0
57	1	1	1	1	1	1	1	0	1	0	0	0	1	1	1	1	0	0
58	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
59	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
60	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
61	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Fig.8 Snapshot of Mandelbrot fractal obtained after conditional formatting

Step 14: Reduce the cell Size. Select columns A to BJ, right click the mouse and choose Column width. Give column width as 0.1. Select Rows A12 to 73, right click the mouse and choose Row height. Give Row-height as 1. The output looks like this (Fig.9).

Step 15: Select Cells B12:BJ72 and go for surface plot. From the plot remove unwanted objects like axis, grids and labels. Select the plot area, Right click the mouse and go for 3D rotation. For

rotation about x axis as 0 degree, rotation about y-axis as 90 degree and perspective as 20 degree, the plot look as in Fig.10

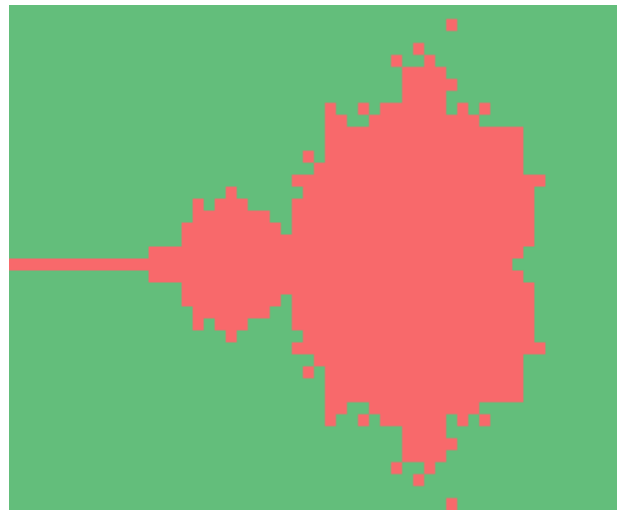


Fig.9 Mandelbrot fractal by conditional formatting

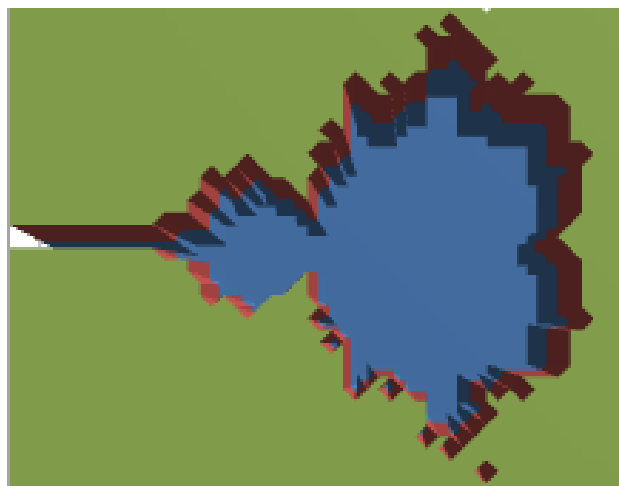


Fig.10 Mandelbrot fractal by surface plot

Experiment No: 2 Get surface plot of Mandelbrot set with color-index given to a cell is same as number of iteration taken to diverge.

In the excel sheet where experiment No: 1 is done, make the following changes

Step 1: In the cell A11, enter =B7. Note that B7 contain the formula for getting count of number iteration taken to diverge. Excel automatically updates the table values. No need of doing what if analysis again!

Step 2: Select the cells from B12:BJ72 and go for surface plot. Remove labels, axes, grids etc. from the plot object. Placing the mouse pointer inside the plot area and right click the mouse and choose 3D rotation. A figure like this can be easily obtained.

	A	B	C
1	n	0	1
2	Cx	0.1	
3	Cy	0.1	
4	xn	0.0	0.1
5	yn	0.0	0.1
6	Diverged?		No
7	Count	32	
8	Color	0	
9			
10			
11	32	-2.00	-1.95
12	-2	0	0
13	-1.95	0	0
14	-1.9	0	0

Fig.11 Snapshot of excel sheet.

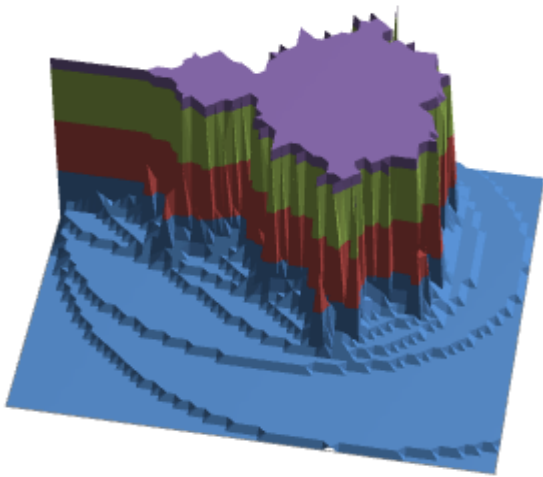


Fig.12 Elevated Mandelbrot set.

Experiment No: 3 Get surface plot (wireframe contour) of Mandelbrot set with color-index given to a cell is by the formula $(n \bmod 8)$ where n is the number of iteration taken to diverge.

In the excel sheet where experiment No: 1 is done, make the following changes

Step 1: In the cell A11, enter $=\text{mod}(B7,8)$

Step 2: Select data and go for surface plot. Within surface plot, there are four options. Choose the last option.

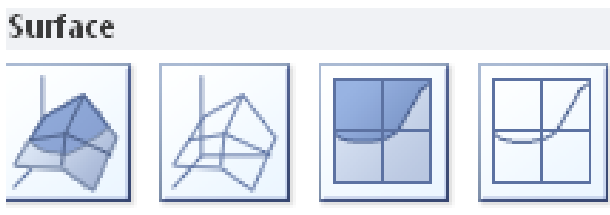


Fig.13 Surface plot options.

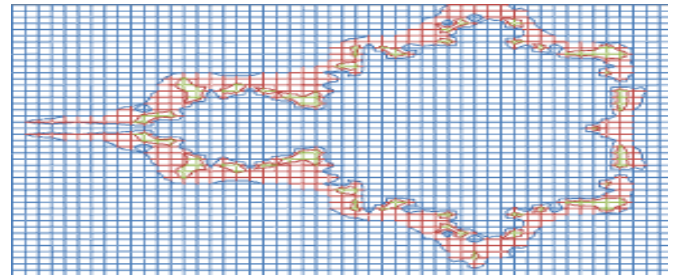


Fig.14 Wireframe contour plot

3. THE JULIA SET

One cannot study the Mandelbrot set without hearing about the Julia set as well. Almost always when the Mandelbrot set is discussed or explained, the Julia set is also mentioned. This document is no exception. Julia sets are named after Gaston Julia, the French mathematician who discovered them and first explored their properties. He lived from 1893 to 1978 and his masterpiece on these sets was published in 1918 (Julia).

So what is the Julia set and what is its relation to the Mandelbrot set?

Many people have seen Julia sets like as shown in Fig.15

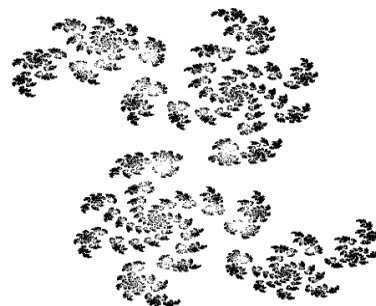
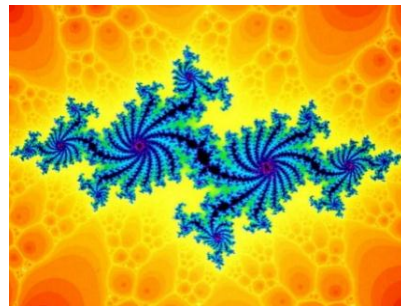


Fig.15 Julia sets

Julia sets can be quite pretty, but their connection to the Mandelbrot set is not clear.

The first connection is the mathematical definition of the set. It's almost identical to the Mandelbrot set:

$$J = \left\{ c \in \mathbb{C} \mid \lim_{n \rightarrow \infty} Z_n \neq \infty \right\}$$

where $z_0 = c = c_x + ic_y$

$$z_{n+1} = z_n^2 + K$$

The only difference to the Mandelbrot set is that the generic z_n function is not " $z_{n+1} = z_n^2 + c$ ", but " $z_{n+1} = z_n^2 + K$ ", where K is some chosen (constant) complex number.

This means that although there is only one Mandelbrot set, there are many Julia sets (actually an infinite amount), one for each possible K (i.e. one for each complex number).

So what's the connection between the Julia set and the Mandelbrot set besides their formulas being almost identical? The value of K has a quite interesting relation with the Mandelbrot set

- If K is inside the Mandelbrot set, then the Julia set for K will be connected. If K is outside the Mandelbrot set, then the Julia set for K will be disconnected (that is, it will have at least two disconnected "islands").
- If a K is chosen from inside the Mandelbrot set, its location inside the set will have a radical influence in the shape of the respective Julia set. The closer K is to the border of the Mandelbrot set, the "thinner" and more "whirly" the corresponding Julia set will be. If we choose a K farther away from the border, the respective Julia set will be "thicker".
- If we choose a K which is very close to the border of the Mandelbrot set, there will also be a close relation between the shape of the Julia set and the shape of the border of the Mandelbrot set at that place. Zooming in the Mandelbrot set around K will bring up shapes that look like the Julia set for K .

For these reasons it is said that the Mandelbrot set is a "map" of all the Julia sets.

A. Plotting Julia set in Excel

Experiment No: 4 Julia set requires the same type of computation as Mandelbrot set. A slight modification in the Excel sheet where the calculation for Mandelbrot set is done will draw the Julia set. For convenience the excel sheet is shown again in Figure 11.5a. In Julia set, complex number K assumes the role of complex number c in the iterative equation for Mandelbrot set. Therefore

Step 1: in cell B2 and B3 enter real and imaginary part of K .

Step 2: in cell B4 and B5, enter any starting values. The main

change is in step 3

Step 3: Select the data table (A11: B17) and go for What-If analysis. It will ask for Row Input Cell and Column input cell. For Julia set, cell B4 is given as Row input cell and B5 as the column input cell. See Fig.16. Note that for Mandelbrot set, cell B2 and B3 are given as the row and column input cell respectively. Rest of the process is same. See Fig.17.

	A	B	C
1	n	0	1
2	Cx	-0.39249	
3	Cy	-0.58797	
4	xn	0.3	-0.5
5	yn	0.4	-0.3
6	Diverged?		No
7	Count	32	
8	Color	0	
9			
10			
11	32	-2.00	-1.95
12	-2	0	0
13	-1.95	0	0

Fig.16 Excel data sheet

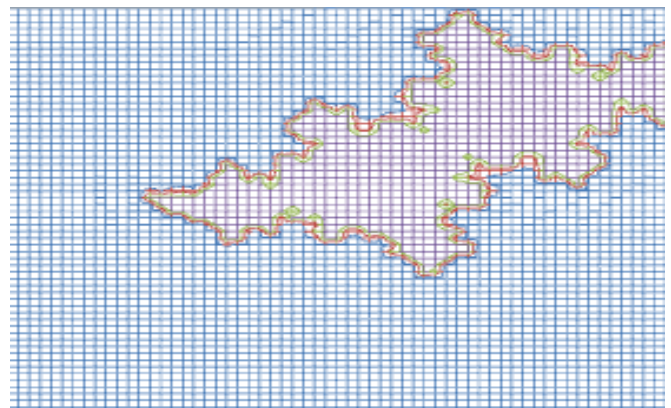


Fig.17 Wireframe contour plot of Julia set

General high resolution implementation using complex number formulas in excel-fractals with transcendental functions and VBA

In Mandelbrot set we used the recursion formula $Z_{n+1} = Z_n^2 + C$ with $Z_0 = 0 + 0i$. Instead of Z_n^2 we can use any higher power of Z . We have a class of functions called Transcendental functions which by Taylor expansion theorem have infinite terms with increasing power of the argument.

For example we have

$$e^z = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \dots$$

$$\sin(z) = z - \frac{z^3}{3!} + \frac{z^5}{5!} - \frac{z^7}{7!} + \dots$$

$$\cos(z) = 1 - \frac{z^2}{2!} + \frac{z^4}{4!} - \frac{z^6}{6!} + \dots$$

$$\sinh(z) = z + \frac{z^3}{3!} + \frac{z^5}{5!} + \frac{z^7}{7!} + \dots$$

$$\cosh(z) = 1 + \frac{z^2}{2!} + \frac{z^4}{4!} + \frac{z^6}{6!} + \dots$$

Many fractal displays are possible by iterating Transcendental functions. The algorithm is same as that of Mandelbrot set. Using VBA in Excel we can produce high quality fractal sets since VBA allows us to have our own colour pallets. A simple VBA code for generating multi coloured fractals in excel spreadsheet is given below.

A. The Cos set

The recursive relation is represented as simplification of cos function value of a complex number in terms of real and imaginary parts and the usually assumed range of domain values. The VBA function code is also given below

$$z_n = \cos(z_{n-1}) + C$$

$$\cos(x + iy) = \cos(x) \cosh(y) - i \cdot \sin(x) \sinh(y)$$

Xmin = -5 ; Xmax = 5 ; Ymin = -8 ; Ymax = +8

Basic code is as follows

```
xpix = 600 ; ypix = 600; max_size = 8; xmax = 12; ymax = 12;
xmin = -12
ymin = -12; maxit = 64 ; deltaq = (xmax - xmin) / xpix;
deltaq = (ymax - ymin) / ypix
For i = 0 To xpix
For j = 0 To ypix
cx = xmin + i * deltaq
cy = ymax - j * deltaq
color = 0
x = 0
y = 0
While ((color < maxit) And (Sqr(x * x + y * y) < max_size))
xn = x
x = cx + Cos(x) * (Exp(y) + Exp(-y)) / 2#
y = cy - Sin(xn) * (Exp(y) - Exp(-y)) / 2#
color = color + 1
Wend
If (color >= maxit) Then
color = 1
Else
color = color Mod 6 + 2
Range("A1").Select
Selection.Cells.Offset(i, j).Interior.ColorIndex = color
End If
Next
Next
```



Fig.18 Cos set

B. The Sin set

$$z_n = \sin(z_{n-1}) + C$$

$$\sin(x + iy) = \sin(x) \cosh(y) + i \cdot \cos(x) \sinh(y)$$

Xmin = -5 ; Xmax = 5 ; Ymin = -8 ; Ymax = +8

C. The Exponential set

$$z_n = C \cdot \exp(z_{n-1})$$

Xmin = -5 ; Xmax = 5 ; Ymin = -8 ; Ymax = +8

D. The Sinh set

$$z_n = \sinh(z_{n-1}) + C$$

Xmin = -5 ; Xmax = 5 ; Ymin = -8 ; Ymax = +8

E. The Cosh set

$$z_n = \cosh(z_{n-1}) + C$$

Xmin = -5 ; Xmax = 5 ; Ymin = -8 ; Ymax = +8

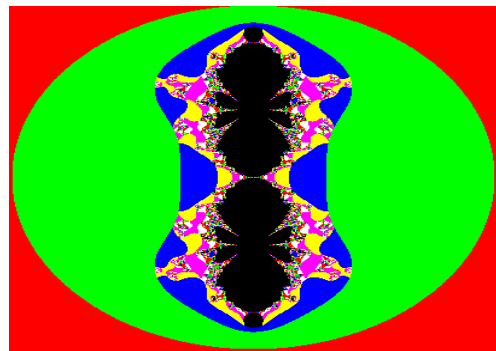


Fig.19 Sin set

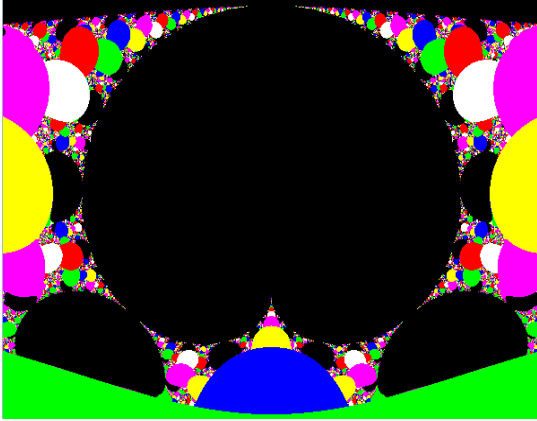


Fig.20 Exponential set

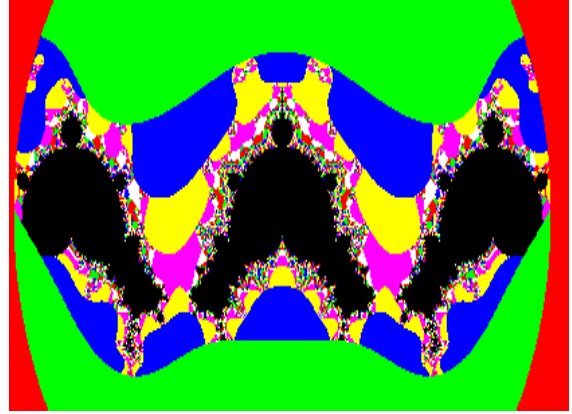


Fig.21 Cosh set

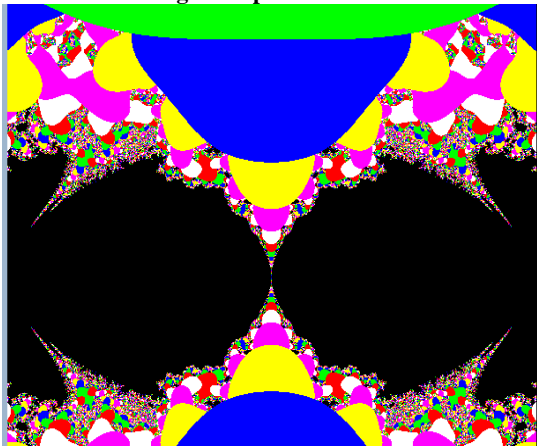


Fig.21 Sinh set

4. SUMMARY

Mandelbrot set is a collection of points in the complex-number plane. Mandelbrot demonstrated the visual beauty of mathematical formula and thus made mathematics more approachable and interesting. What Mandelbrot did is close to miraculous as attested by millions of posters, greetings cards and T-shirts, bought by people who had not the faintest idea of what it was. In this article it is shown how to draw it in Excel and explore further. The related sets such as Julia sets and sets based on transcendental functions are also discussed.

REFERENCES

- [1] GNU Xaos: Fastinteractive fractal zoomer, <http://wmi.math.u-szeged.hu/xaos/doku.php>. Accessed 6 August 2012
- [2] Mandelbrot Images, <http://warp.povusers.org/snaps/fract/>. Accessed 6 August 2012.
- [3] Fractintfreeware fractal generator, <http://spanky.triumf.ca/www/fractint/fractint.html>. Accessed 6 August 2012 .
- [4] 3D Mandelbrot set, <http://www.skytopia.com/project/fractal/2mandelbulb.html>. Accessed 7 August 2012.
- [5] Source code for many fractal programs, <https://www.fractalus.com/downloads/source/>. Accessed 4 August 2012.