U3A Maths

15-Sep-2026 Term 3 / Week 9

Complex Function Plots - Julia Set

Review

- Till now, we have considered the following methods of plotting complex functions

: 3-dimensional plots

- Plotting of complex functions requires 4 dimensions. However, it is possible to plot one of the outputs, for Eg. Magnitude, on a three dimensional plot.

- Such 3-dimensional plots result in spectacular plots, especially in colour!
- The most spectacular plot was produced by plotting the Sombrero function, namely,

$$f(z) = \frac{\sin(z)}{z}$$

Domain Coloring Plots

- Domain coloring method plots the complex functions on a complex plane (x-y axis) by assigning "colours" to "angle" and "brightness" to "magnitude".
- Magnitudes are also illustrated with contours.

- This method was developed in the 1990's consequent to the introduction of computers with colour screens.

- Domain coloring is especially well suited for conceptual visualisation of poles and zeros of complex functions.

Iterative Function Plots

- Iterative function plots using domain coloring can produce truly "beautiful" plots using simple functions. These plots can often produce mysterious shapes occuring in nature

- Iterative plots are produced by using output value of the function, as the input for the next iteration/plot.

- For Ex: Consider $f(z) = z^2 + c$

  let $z = z_0$ be the initial value of 'z'.

  1st iteration

  $\therefore z_1 = f(z_0) = z_0^2 + c$

  2nd iteration

  $z_2 = f(z_1) = z_1^2 + c$ etc

- We could produce designs of oriental carpets by iterating $f(z) = \dfrac{\cos(z)}{\sin(z^4 - 1)}$

-5-

- For interactive plotting of the above function, visit the web site "peterfrancis.com/complex-function-plot/"
- A variation of the iterative plotting of complex functions using domain coloring can produce mindboggling plots.
- The most famous of such plots are:

Julia set - developed by Gaston Julia (1893-1978) (French) - one of the main inventors of Fractals & iterative plots.

-6-

Mandelbrot set - developed by Benoit Mandelbrot (1924-2010) (Polish/French/American) - another major contributor to "Fractal Geometry".

- We will go through Julia set in detail in this class, to illustrate the concepts.
- Mandelbrot set is essentially a generalised version of Julia set!

-7-

Julia set

- Julia set is obtained by iterative process of the function:

$$f(z) = z^2 + c$$

where 'c' is a given constant.

- It is interesting to note that the function values themselves are not plotted!!
- Instead, at each (initial) 'z' value on the complex x-y axes, the number of iterations of the function are stored.

-8-

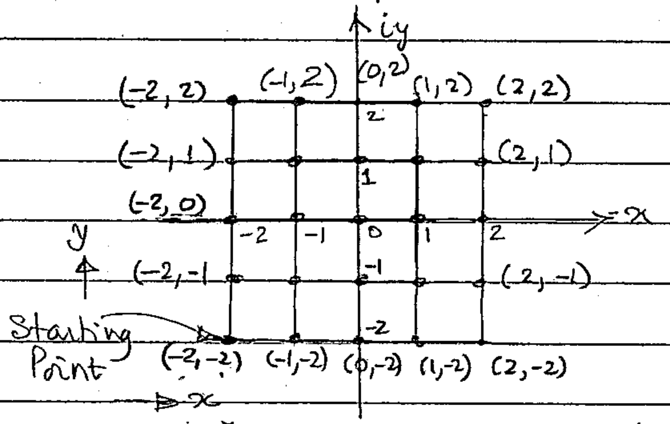
- The function iteration at a 'z' value is stopped when the $|f(z)|$ exceeds a specified boundary. For Ex, iteration is stopped when $|f(z)| \geq 2$
- The resulting set (matrix) containing the "number of iterations" is the Julia set.
- We can plot such a set!
- Let us consider a simple example to illustrate the concepts.
- First let us choose

$$c = -0.7 + i\,0.27$$

& $|f(z)| \geq 2$ iteration stops

let Max. No. of iterations = 5

Ex: Calculate the Julia set for the following 'Z_0' values (5x5) on the complex plane.



Expressing above in matrix form

$$[Z_0] = \begin{array}{c|ccccc} y \downarrow \; x \rightarrow & -2 & -1 & 0 & 1 & 2 \\ \hline -2 & (-2-2i) & (-1-2i) & (0-2i) & (1-2i) & (2-2i) \\ -1 & (-2-1i) & (-1-1i) & (0-1i) & (1-1i) & (2-1i) \\ 0 & (-2+0i) & (-1+0i) & (0+0i) & (1+0i) & (2+0i) \\ 1 & (-2+1i) & (-1+1i) & (0+1i) & (1+1i) & (2+1i) \\ 2 & (-2+2i) & (-1+2i) & (0+2i) & (1+2i) & (2+2i) \end{array}$$

- To simplify our calculations, the resulting Julia set (ie, the number of iterations) is as given below:

$$[J] = \begin{array}{c|ccccc} y \downarrow \; x \rightarrow & -2 & -1 & 0 & 1 & 2 \\ \hline -2 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 2 & 2 & 0 \\ 0 & 1 & 5 & 5 & 5 & 1 \\ 1 & 0 & 2 & 2 & 1 & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 \end{array}$$

- Let us check the calculations for $Z_0 = (0 - 1i)$
  These values are circled on the above matrices $[Z_0]$ & $[J]$
- As shown, the number of iterations = 2
  Let us check.

Iteration # 0 $\quad Z_0 = (0 - 1i)$

we have $|Z_0| = 1$ ∴ less than 2.
∴ continue iteration.

Iteration # 1

$$Z_1 = (Z_0)^2 + C$$
$$= (0-1i)^2 + (-0.7+0.27i)$$
$$= (-1.7 + 0.27i)$$

we have $|Z_1| = 1.721$ less than 2
∴ continue iteration

Iteration #2

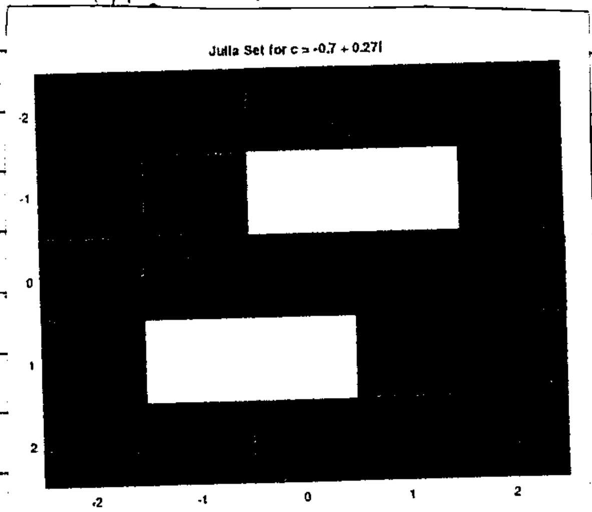
$$Z_2 = Z_1^2 + C$$
$$= (-1.7+0.27i)^2 + (-0.7+0.27i)$$
$$= (2.11 - 0.648i)$$

We have $|Z_2| = 2.21$ is greater than 2

- Iteration stops here and the No. of iterations = 2 is recorded.

Homework Check results for $Z_0 = (-1+1i)$

- We can now plot Julia set (J matrix) by assigning colours, say 0 1 2 3 4 5
  Blue .... White .... Red
- A typical plot in B/W is as below



- The above plot looks trivial!
- However, we can obtain mind boggling plots for large values points and iterations. We will check it Next Week!

```
%
% Julia set
%
clc;
output_precision(3);

% Image resolution
    width  = 701;     % Start with 5, 21 ... 901
    height = 701;     % Start with 5, 21 ... 901

% Complex plane boundaries
    x = linspace(-2, 2, width);
    y = linspace(-2, 2, height);
    [X, Y] = meshgrid(x, y);
    Z = X + 1i * Y;
    Z0 = Z;         % Initial Z matrix for Display

% Constant for the Julia set (See 'End of Code' below for values of 'c')
    c = -0.8 + 0.156i ;     % Original  (-0.7 + 0.27015i) => Galaxy

% Maximum number of iterations
    max_iter = 500;     % Try 5,100,200,300,400,500 to illustrate resolution
    output = zeros(size(Z));

% Iterate Zn+1 = Zn^2 + c  (Initial value is Z0)
    for i = 1:max_iter
        mask = abs(Z) <= 2;
        Z(mask) = Z(mask).^2 + c;
        output(mask) = i;
    end

% Render the fractal
    imagesc(x,y,output);
    colormap(jet);        % Try 'hot', 'turbo', 'jet'
    % axis off;
    title(['Julia Set for c = ', num2str(real(c)), ' + ', num2str(imag(c)), 'i']);


%{
   c = 0                 circle
   c = -2                needle like straight line
   c = -0.4 + 0.6i       intricate dendriitic structure
   c = -0.8 + 0.156i     famous dendrite or San Marco structure
   c = -0.7 + 0.27i      famous spiral galaxy
   c = 0.285 + 0.01i     island like archipelago shape
   c = -0.70176-0.3842i  Organic shape - biological cells
%}
```