

Mandelbrot Set

Julia Set Review

- Julia set uses the iterative equation:

$$Z_{p+1} = Z_p^2 + c$$

Where 'c' is a given complex constant

The starting value ($p=0$):

$$Z_0 = Z_{xy} \quad (x=1..m \text{ \& } y=1..n)$$

where Z_{xy} are the points on the complex plane. Hence, the resulting ' Z_p ' values is a matrix of size $(m \times n)$.

- The calculations using the above equation for each Z_{xy} is done till $|Z| \leq 2$.

-3-

Mandelbrot Set

- Mandelbrot uses the same equation, namely

$$Z_{p+1} = Z_p^2 + c$$

but, with a slight variation.

- Instead of a user specified value 'c', the value of c is set as Z_{xy}

$$\text{ie, } c = Z_{xy}$$

where Z_{xy} are the points on the complex x-y plane.

- The calculations are done for each point (Z_{xy}) on the x-y plane, that is $(m \times n)$ points.

-2-

- Julia matrix consists of the number of iterations for each Z_{xy} value

- Hence, Julia set is a matrix of integers! This matrix is plotted in the (x,y) plane as per the chosen colour scheme.

- obviously, varying the 'c' value will generate a different set and a corresponding plot.

- Julia sets result in mind-boggling plots, for ex:
 $c = (-0.7 + i0.2) \Rightarrow$ spiral galaxy
 $c = (-0.8 + i0.16) \Rightarrow$ Dendrites (San Marco)
 etc

-4-

- The initial value of Z_p is always set to zero.
 that is, $Z_0 = 0$

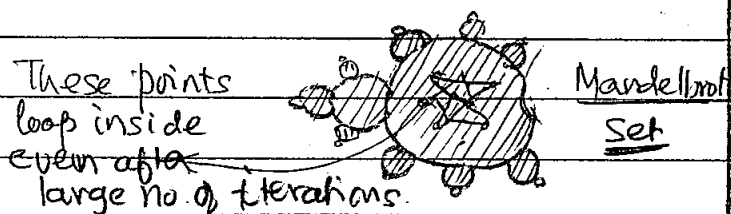
- Calculations are the same as the Julia set.

The number of iterations are stored as a matrix and such a matrix is called the Mandelbrot matrix.

- Mandelbrot was initially interested only in the points which reached the maximum number of iterations and still remained within $|Z| \leq 2$. It is called the 'Mandelbrot set'

• When plotted in black & white (using a printer), he was intrigued by the shape of the set. The set had repeated shapes with smaller sizes. He called them "Fractals".

• Mandelbrot set essentially replicates natural evolution. The same basic structure evolving into larger structures. A simple illustration is as below:



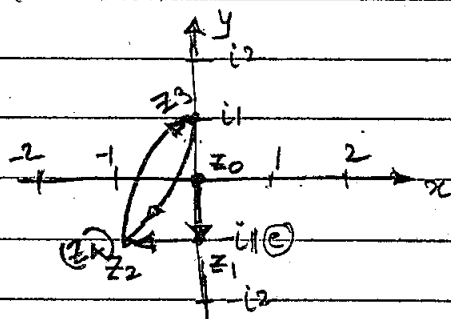
$$z_2 = z_1^2 + c = (0-i)^2 + (0-i) = (-1-i)$$

$$z_3 = z_2^2 + c = (-1-i)^2 + (0-i) = (0+i)$$

$$z_4 = z_3^2 + c = (0+i)^2 + (0-i) = (-1-i)$$

z_4 is same as z_2 , hence the values will loop. Hence,

$c = (0-i)$ is in the Mandelbrot set.



• With the modern computer plots, the points outside the Mandelbrot set is plotted with various colour schemes, by assigning colours to the number of iterations.

• The above structures in the Mandelbrot set evolve infinitely.

• Finally, it is important to note that, for a given point inside the Mandelbrot set, the ' z ' values during iterations will loop within the boundary $|z| \leq 2$.

For Ex, let $C = z_{xy} = (0-i)$

check whether this value is in the Mandelbrot set

We have, $z_0 = 0$

$$z_1 = z_0^2 + c = 0^2 + (0-i) = (0-i)$$

• The resulting plot is truly mind boggling.

• In fact, Julia set plots are a subset of Mandelbrot plot, since the ' c ' value used in Julia set is a part of the Mandelbrot calculations.

• Visit the website "www.fractalset.com/Julia" to explore the Julia and the Mandelbrot set in detail.

Homework

check if $c = (-1-i)$ is within Mandelbrot set.

```

%
% Mandelbrot Plot - the Vector (matrix) method !!! - Very fast
%
grid_size = 701;    % Resolution (Also try 5x5 and 801x801 pixels)
max_iter = 100;     % Maximum number of iterations

% Create the complex plane grid (c)
x = linspace(-2.0, 2.0, grid_size); % original values -2 to 0.5
y = linspace(-2.0, 2.0, grid_size); %Original values -1.25 to 1.25
[X, Y] = meshgrid(x, y);
C = X + 1i * Y;      % 'C' is a matrix of constants, that is Zxy values

% Initialize variables
Z = zeros(size(C));
M = zeros(size(C)); % Matrix to store iteration counts

% Vectorised Mandelbrot Iterations
% Vectorised equations update the whole matrix with a single statement!!
for k = 1:max_iter
    % 'mask' is a matrix of logic values; set to 1 if Z<=2 & set to 0 if not True
    mask = (abs(Z) <= 2);

    % Z matrix values are only updated only if the corressponding 'mask' value is 1
    Z(mask) = Z(mask).^2 + C(mask);

    % 'M' matrix stores the number of iterations
    % No of iterations are updated only if the corresponding 'mask' value is 1
    M(mask) = k;
end

% Plotting the Mandelbrot set
figure;
imagesc(x, y, M); % Display the iteration matrix as an image
colormap(jet);    % Set the color scheme (try 'hot', 'ocean', or 'prism')
%colorbar;        % Add a color scale legend
%axis off;        % Hide the axis numbers
title ('Mandelbrot Set');

```

