

```

%generates a CA using complex numbers

%generates a random 4-bit initial vector

clear all

num_rows = 250;
num_iterations = 500;

initial_value = 1;
boundary_condition = -i;

initial_conditions = ones(num_rows,1)*initial_value;

%replaces values with states in [1,-1,i,-i]
x = find(initial_conditions == 2);
initial_conditions(x) = -1;
x = find(initial_conditions == 3);
initial_conditions(x) = i;
x = find(initial_conditions == 4);
initial_conditions(x) = -i;

[auto_matrix] = Generate_complex_CA(initial_conditions, boundary_condition, num_rows, num_iterations);

%generates RGB image where cells are colored white, black, green, blue, respectively over [1,-1,i,-i]

Display_MatrixR = zeros(num_rows,num_iterations);
Display_MatrixG = zeros(num_rows,num_iterations);
Display_MatrixB = zeros(num_rows,num_iterations);

%white cells
x = find(auto_matrix == 1);
Display_MatrixR(x) = 255;
Display_MatrixG(x) = 255;
Display_MatrixB(x) = 255;

%black cells
x = find(auto_matrix == -1);
Display_MatrixR(x) = 0;
Display_MatrixG(x) = 0;
Display_MatrixB(x) = 0;

%green cells
x = find(auto_matrix == i);
Display_MatrixR(x) = 0;
Display_MatrixG(x) = 255;
Display_MatrixB(x) = 0;

%blue cells
x = find(auto_matrix == -i);
Display_MatrixR(x) = 0;
Display_MatrixG(x) = 0;
Display_MatrixB(x) = 255;

Display_Matrix(:,1) = Display_MatrixR;
Display_Matrix(:,2) = Display_MatrixG;
Display_Matrix(:,3) = Display_MatrixB;

Display_Matrix = Display_Matrix / 255;

Display_Matrix(1,1,:) = 1;
Display_Matrix(num_rows,1,:) = 1;

figure, imshow(Display_Matrix)

```