

FRACTAL METHODS IN COMPUTER GRAPHICS

Assignments set 3

In all the assignments, we need to use colour maps. Examples of colour maps can be found in the *colormaps.zip* archive. In the archive, each colour map is in two formats: (1) an image that can be used as a 1D texture, (2) a text file in which, in the successive lines, we have the R, G, B values of the colours that form the map.

1. Implement the algorithm for generating the Mandelbrot set (Algorithm 1).

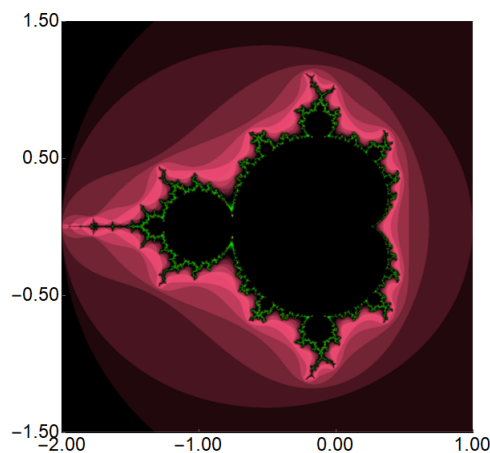
Algorithm 1: Mandelbrot set

Input: $K \in \mathbb{N}$ – the maximal number of iterations; $A \subset \mathbb{C}$ – area.

Output: Approximation of the Mandelbrot set in A .

```
1 for  $c \in A$  do
2    $R = \max\{|c|, 2\}$ 
3    $z = c$ 
4    $i_c = 0$ 
5   for  $i = 1$  to  $K$  do
6      $z = z^2 + c$ 
7     if  $|z| > R$  then
8        $i_c = i$ 
9       break
10  Colour the point  $c$  with the  $i_c$  colour
```

Example of the Mandelbrot set for $K = 50$, $A = [-2, 1] \times [-1.5, 1.5]$ and the *neon.map* colour map:



2. Implement Julia set generation algorithm in which the colouring is done with the help of the binary decomposition (Algorithm 2).

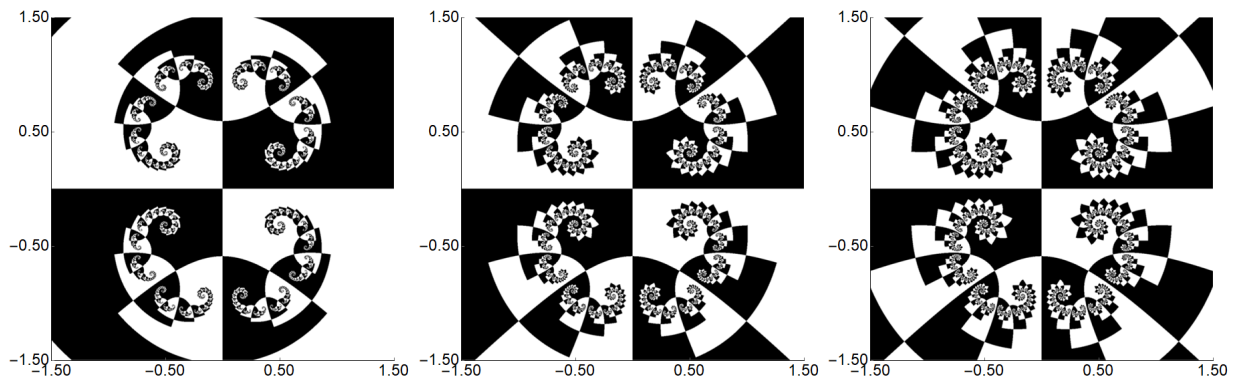
Examples of Julia sets for $c = 0.35$, $A = [-1.5, 1.5]^2$, $K = 50$ and various values of the escape threshold R , from left 2, 5, 10:

Algorithm 2: Julia set.

Input: $c \in \mathbb{C}$ – the parameter; $A \subset \mathbb{C}$ – area; $K \in \mathbb{N}$ – the maximal number of iterations; $R \in \mathbb{R}_+$ – the escape threshold.

Output: Approximation of the Julia set in A .

```
1 for  $z_0 \in A$  do
2    $i_c = 0$ 
3   for  $i = 1$  to  $K$  do
4      $z_i = z_{i-1}^2 + c$ 
5     if  $|z_i| > R$  then
6        $i_c = i$ 
7       break
8   Colour the point  $z_0$  using the binary decomposition for  $z_i$ 
```



3. Implement polynomiographs generation algorithm with iterations (Algorithm 3). In the program, we should have the possibility of selecting: the root-finding method (at least two methods, both other than Newton's method), the polynomial, iteration type (at least two iterations, both other than Picard's iteration), and the convergence test (at least two different tests).

Examples of root-finding methods, that can be used in the program, can be found, for instance, in

- Section 5 „Newton Fraktale“ <http://www.3d-meier.de/tut20/Seite1.html>
- Gościński, I., Gdawiec, K.: PSO-based Newton-like Method and Iteration Processes in the Generation of Artistic Patterns. Lecture Notes in Computer Science, vol. 11241, pp. 47-56, (2018) [PDF version](#)
- Gdawiec, K., Kotarski, W., Lisowska, A.: On the Robust Newton's Method with the Mann Iteration and the Artistic Patterns from Its Dynamics. Nonlinear Dynamics 104(1), 297-331, (2021) [PDF version](#)

A review of 17 different iterations can be found here:

- Gdawiec, K., Kotarski, W.: Polynomiography for the Polynomial Infinity Norm via Kalantari's Formula and Nonstandard Iterations. Applied Mathematics and Computation 307, 17-30, (2017) [PDF version](#)

Algorithm 3: Polynomiograph rendering.

Input: $p \in \mathbb{C}[Z]$, $\deg p \geq 2$ – polynomial; $A \subset \mathbb{C}$ – area; M – the number of iterations; $I_v : \mathbb{C} \rightarrow \mathbb{C}$ – the iteration with the parameters v ; $C_u : \mathbb{C} \times \mathbb{C} \rightarrow \{true, false\}$ – convergence test; $colours[0..k]$ – colour map.

Output: Polynomiograph in A .

```

1 for  $z_0 \in A$  do
2    $[n, z] = \text{ITERATEPOINT}(z_0, p, I_v, C_u, M)$ 
3   Colour  $z_0$  using  $n, z$  and the colour map  $colours$ 

```

Algorithm 4: ITERATEPOINT

Input: $z_0 \in \mathbb{C}$ – the point; $p \in \mathbb{C}[Z]$, $\deg p \geq 2$ – polynomial; $I_v : \mathbb{C} \rightarrow \mathbb{C}$ – the iteration with the parameters v ; $C_u : \mathbb{C} \times \mathbb{C} \rightarrow \{true, false\}$ – convergence test; M – the number of iterations.

Output: The number of iteration and the last computed point.

```

1 ITERATEPOINT( $z_0, p, I_v, C_u, M$ )
2    $n = 0$ 
3   while  $n < M$  do
4      $z_{n+1} = I_v(z_n)$ 
5     if  $C_u(z_n, z_{n+1}) = true$  then
6       break
7      $n = n + 1$ 
8   return  $[n, z_{n+1}]$ 

```

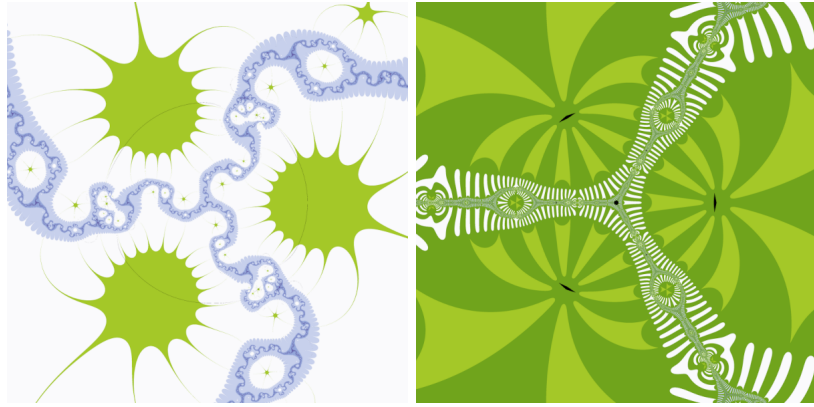
Example 1: polynomiographs for Halley's method, $p(z) = z^3 - 1$, $A = [-2, 2]^2$, $M = 30$, $\varepsilon = 0.001$, the colour map from *0408_093-s.map* and the Picard-S iteration with the parameters:

- $\alpha = 0.5 + 1.5i$, $\beta = 0.8$ and the convergence test

$$||z_{n+1}|^2 - |z_n|^2| < \varepsilon,$$

- $\alpha = 0.1$, $\beta = 0.9$ and the convergence test

$$|0.01(z_{n+1} - z_n)| + |0.029|z_{n+1}|^2 - 0.03|z_n|^2| < \varepsilon.$$



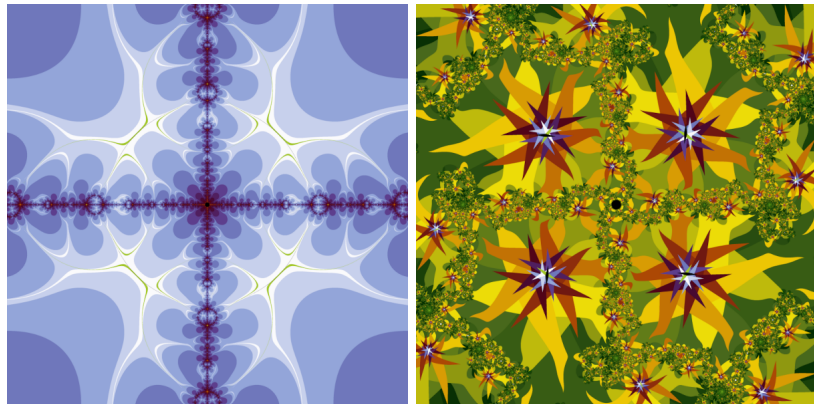
Example 2: polynomiographs for Halley's method, $p(z) = z^4 + 1$, $A = [-2, 2]^2$, $M = 30$, $\varepsilon = 0.001$, the colour map from *0408_093-s.map* and the Ishikawa iteration with the parameters:

- $\alpha = 0.9$, $\beta = 0.1$ and the convergence test

$$||z_{n+1}|^2 - |z_n|^2| < \varepsilon,$$

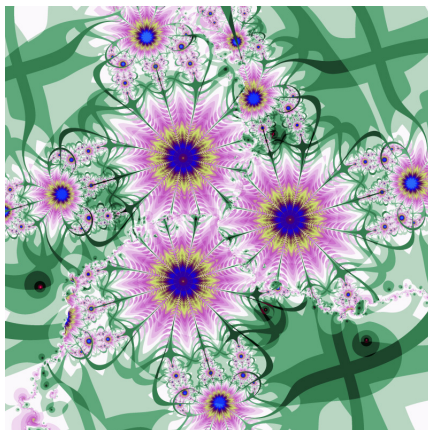
- $\alpha = 0.6 - 0.75i$, $\beta = 0.8$ and the convergence test

$$|0.01(z_{n+1} - z_n)| + |0.029|z_{n+1}|^2 - 0.03|z_n|^2| < \varepsilon.$$



4. Implement the multi-step polynomiography (Algorithm 5).

Multi-step polynomiograph:



Algorithm 5: Multi-step polynomiograph rendering.

Input: $A \subset \mathbb{C}$ – area; $\{p_1, p_2, \dots, p_N\}$ – polynomials; $\{I_{v_1}, I_{v_2}, \dots, I_{v_N}\}$ – iterations; $\{M_1, M_2, \dots, M_N\}$ – the numbers of iterations for the successive steps; $\{C_{u_1}, C_{u_2}, \dots, C_{u_N}\}$ – convergence tests for the successive steps; $\{f_1, f_2, \dots, f_N\}$ – area transformations for the successive steps; $colours[0..k]$ – colour map.

Output: Multi-step polynomiograph in A .

```
1 for  $z_0 \in A$  do
2    $m = 0$ 
3    $z = z_0$ 
4   for  $i = 1, 2, \dots, N$  do
5      $[n, u] = \text{ITERATEPOINT}(z, p_i, I_{v_i}, C_{u_i}, M_i)$ 
6      $m = m + n$ 
7      $z = f_i(u - z)$ 
8   Determine the colour for  $z_0$  using  $m$  and the colour map  $colours$ 
```

was generated in $[-3, 3]^2$ using the *jutemap.map* colour map, $\varepsilon = 0.001$ and the following parameters:

- the first step: Halley's method, $p_1(z) = z^3 - 1$, $M_1 = 20$, Noor iteration with $\alpha = 0.8 - 0.4\mathbf{i}$, $\beta = 0.2$, $\gamma = 0.9 - 0.5\mathbf{i}$, $f_1(z) = z / (0.75 + 2.5\mathbf{i})$ and the convergence test:

$$|z_{n+1} - z_n| < \varepsilon,$$

- the second step: Newton's method, $p_2(z) = z^4 + 1$, $M_2 = 20$, Ishikawa iteration with $\alpha = 0.7$, $\beta = 0.7$, $f_2(z) = z / (0.75 + 2.5\mathbf{i})$ and the convergence test:

$$|0.04\Re(z_{n+1} - z_n)| < \varepsilon \vee |0.05\Im(z_{n+1} - z_n)| < \varepsilon,$$

where $\Re(z)$, $\Im(z)$ are the real and imaginary parts of the complex number z , respectively.