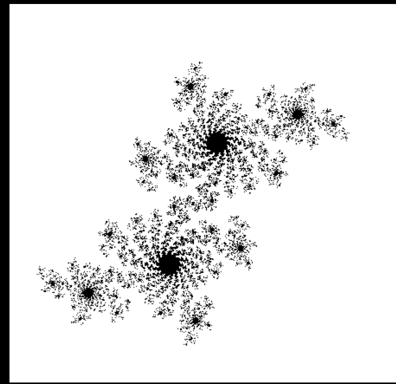
Julia and Mandelbrot sets

I.Kolingerová

References:

- Francis S.Hill Jr.: Computer Graphics, Macmillan Publishing Company, New York, 1990
- H.A. Lauwerier, J.A. Kaandrop: Fractals (Mathematics, Programming and Applications), TR CS-R8762, Centre for Mathematics and Computer Science, Amsterdam, The Nietherlands, 1980
- J.C.Sprott, C.A. Pickover: Automatic Generation of General Quadratic Map Basins, Computers & Graphics, Vol.19, No.2, pp.309-313, 1995

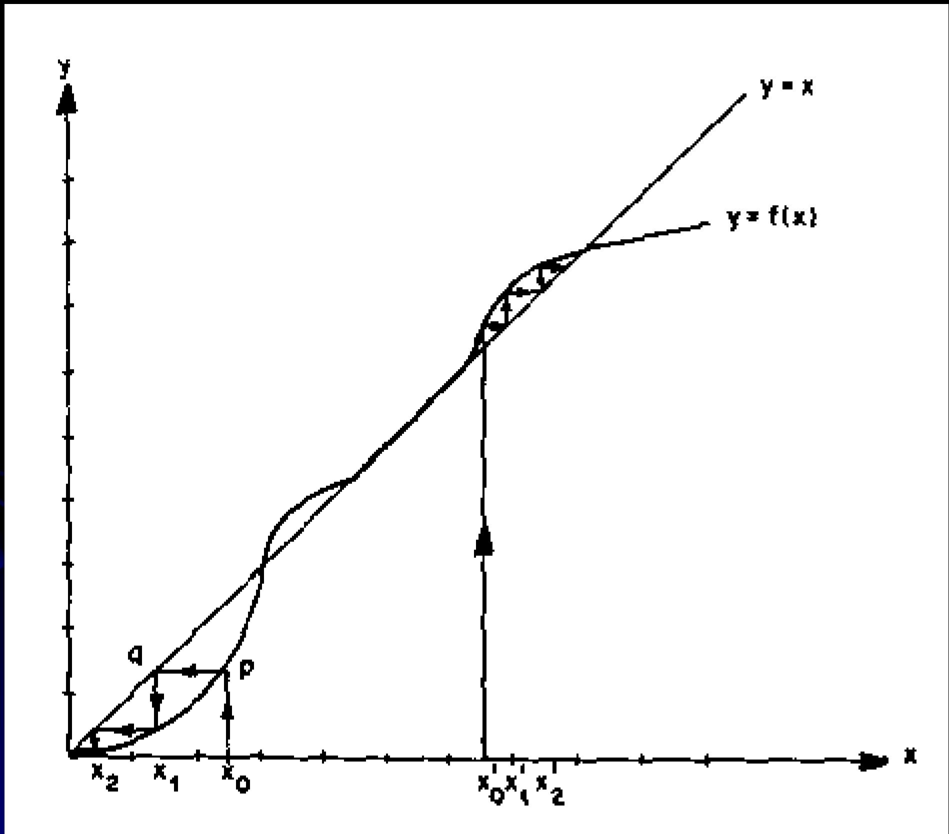
Julia sets



- French mathematician G.Julia, 1918
- J. set: $z_{n+1} = F(z_n)$, z_n – a complex number
- It preserves angles but the scale depends on the z value (locally, it is a rotation with a scale, the scale factor $|F'(z)|$
- Standard example: $z_{n+1} = z_n^2 + c$, $c = a+ib$
  in real numbers: $x_{n+1} = x_n^2 - y_n^2 + a$,
  $$y_{n+1} = 2x_ny_n + b$$
- It is important to inspect fixed and periodic points of F

- Fixed point: given by $z = F(z)$
- If $|F'(z)| < 1$, the point is stable. If z_0 is near a fixed point z, then the orbit z0, $z_1, z_2, z_3, ..$ converges to z. Then z is the attractor.
- If $|F'(z)| > 1$, then the point is unstable, a repellor.
- If $|F'(z)| = 1$, the fixed point is neutral.
- Periodic orbit (m-cycl): $z_m = F(z_{m-1}) = z_0$, m – the smallest integer, for which the equality holds; z_0 – a periodic point of order m. M-cycl is stable if $|F'(z_0)F'(z_1)F'(z_2)...F'(z_{m-1})| < 1$, analogically for unstability.
- Stability and unstability are very useful for systems control, attractors for graphics

Example 1

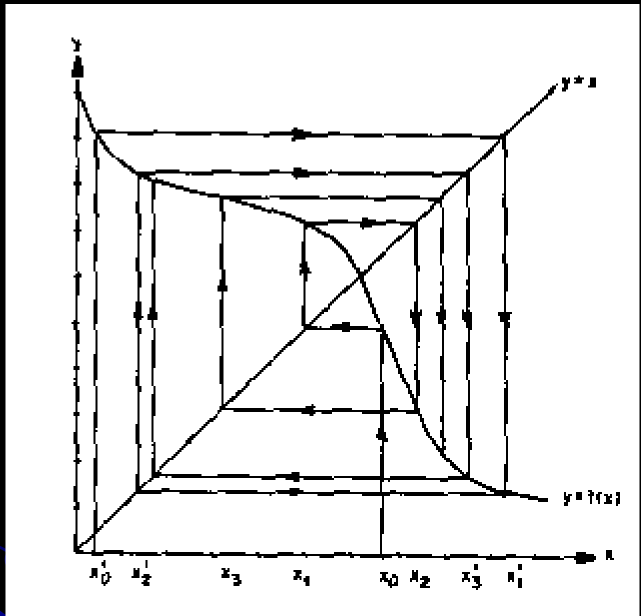


Monotonous function, positive derivation, does not produce chaos

Fixed points are $x^{(1)}=0$, $x^{(2)}=3/10$, $x^{(5)}=4/5$, interval J=1/2 to 3/5

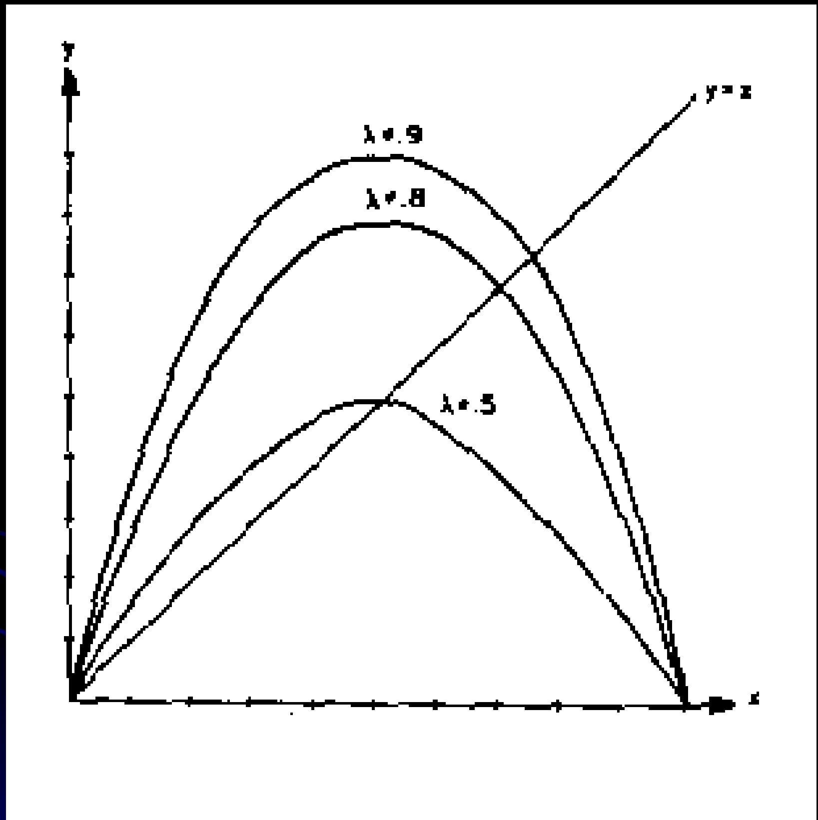
From the starting point x_0, x_0': $x^{(1)}$ and $x^{(5)}$ are attractors, $x^{(2)}$ is a repellor while
J attracts close points on the left and repells close points on the right

Example 2



Monotonous function,
negative derivation,
does not produce chaos

Example 3

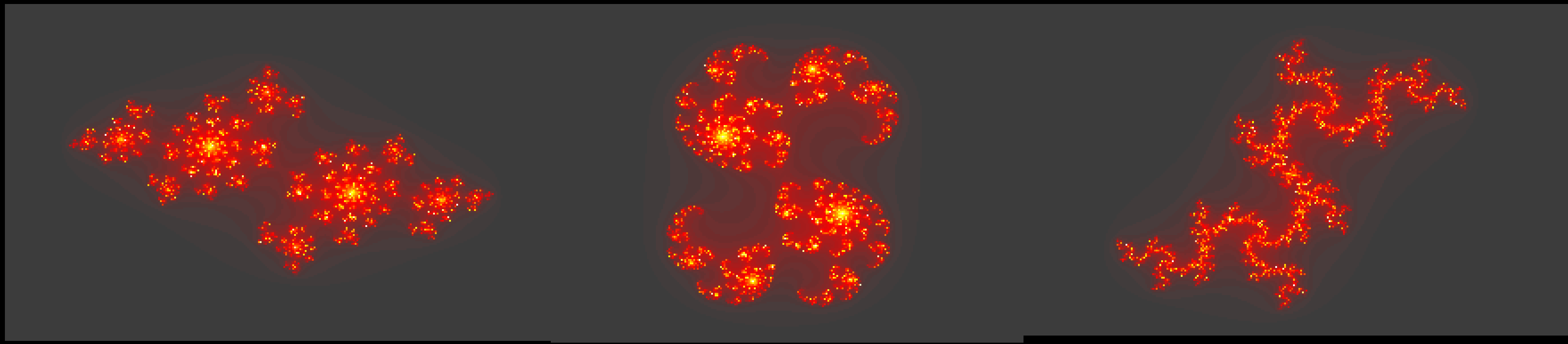


The function f.5: x=0 is a repellor and x=0.5 an attractor,

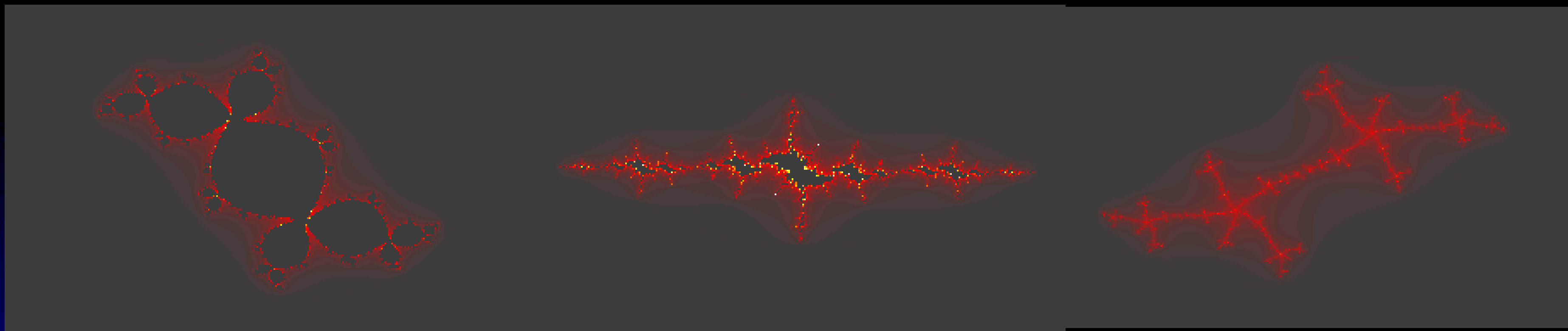
f.8: x=0 is a repellor, x=11/16 is also a repellor and the 2-cycl (x ~ 0.51, x ~ 0.8) is an attractor

f.9 has periodic orbits 2^n and the iterations are chaotic

- Definition: Julia set – a set of all complex numbers z for which the iteration $F(z) \to z^2+c$ je is limited for some c.
- In other words: the graph of all complex numbers z which are not growing to ∞ when they are iterated in $F(z) \to z^2+c$, where c is constant.
- Take a given c, find the color for the pixel (x,y) using $z_0=x+iy$ as a starting point
- More iterations – more details in the drawing
- Julia set is continuous if c lies on the corresponding Mandelbrot set and vice versa (i.e. the orbit for c=0 decides).

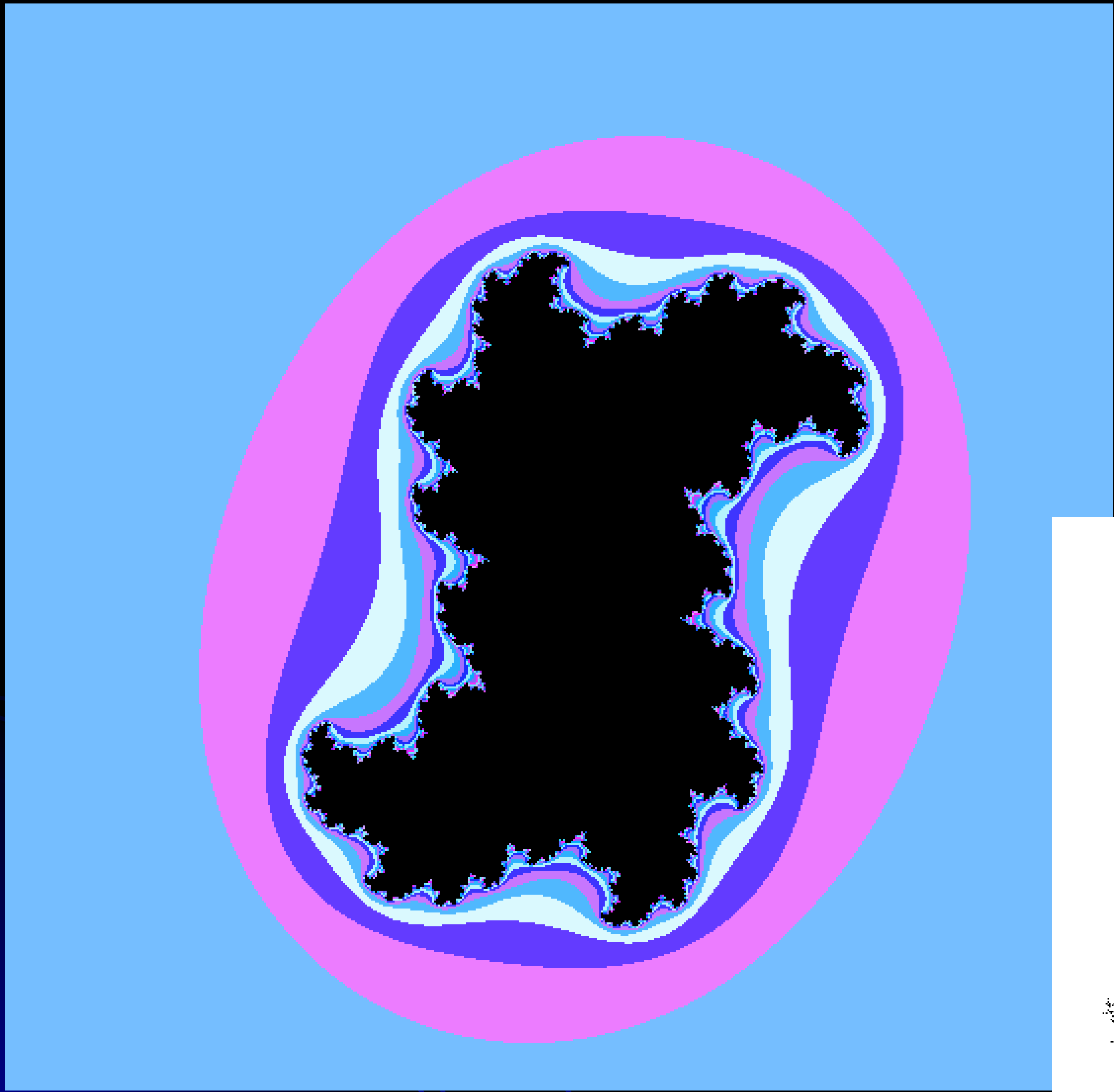


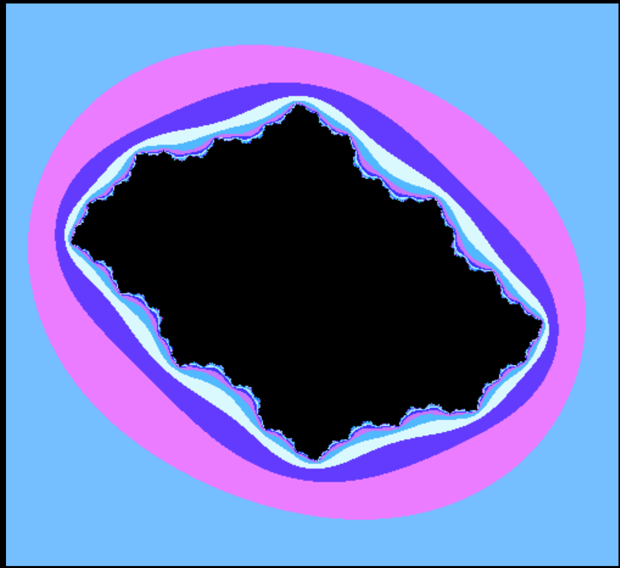
Julia sets for F(z) ->z^2+c

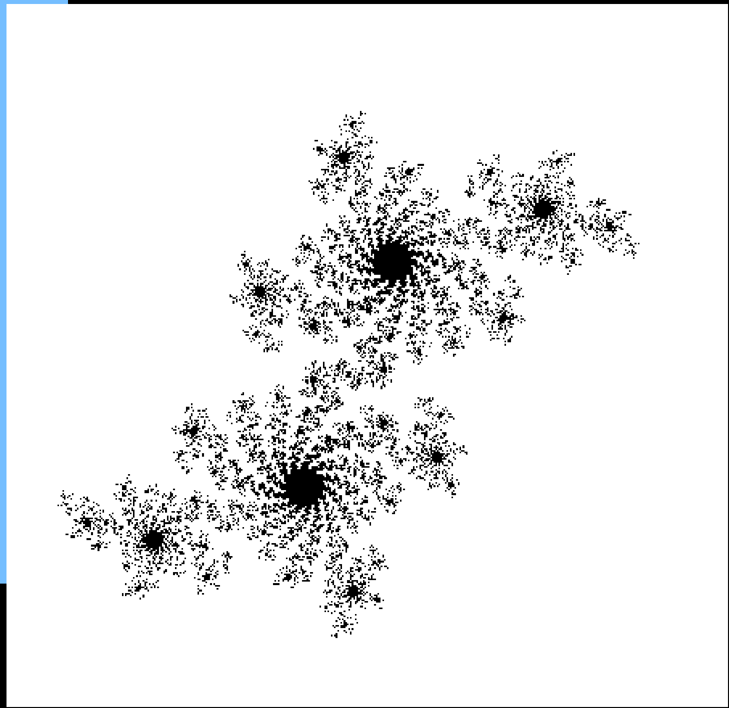


```
function JuliaCount (x,y: extended; num: longint) : longint;

{ num is the maximum number of iterations }

const thresh = 4.0; { a larger threshold may yield better pictures }

var
  cx,cy,tmp,fsq : extended;
  count : longint;

begin
 cx := 0.0005; cy := 0.87;
 fsq := 0;
 count := 0;
 while (count < num) and (fsq <= thresh) do
      begin
       count := count+1;
       tmp := x;
       x := x*x - y*y + cx;
       y := 2.0*tmp*y + cy;
       fsq := x*x + y*y;
      end;
 JuliaCount := count;
end; { JuliaCount }
```

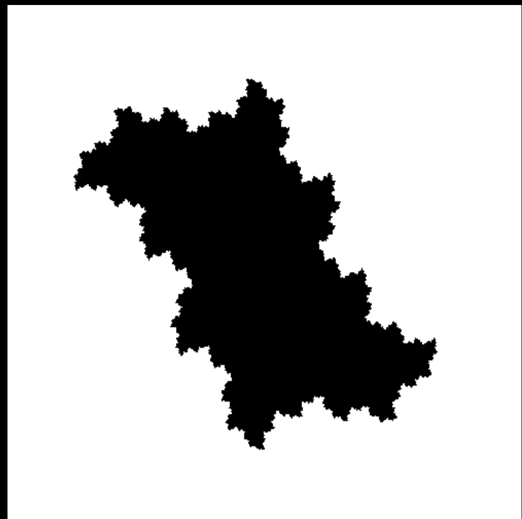
```
procedure Fill_pixels (var cells : TArray);

{ procedura spocte Juliovu mnozinu do pole cells }

var i,j : integer;
    count,num : longint;
    x,y : real;
    col : TColor;

begin
 nrows := Form1.ClientHeight; ncols := Form1.ClientWidth;
 num := 1000;
 for i := 0 to nrows-1 do
     begin
      y := ymin+(ymax-ymin)*i/(nrows-1);
      for j := 0 to ncols-1 do
          begin
           x := xmin+(xmax-xmin)*j/(ncols-1);
           count := JuliaCount (x,y,num);
           if count=num then cells[j,i] := clBlack { point in the Julia set }
           else
             cells[j,i] := clWhite;
          end;
     end
end; { Fill_pixels }
```

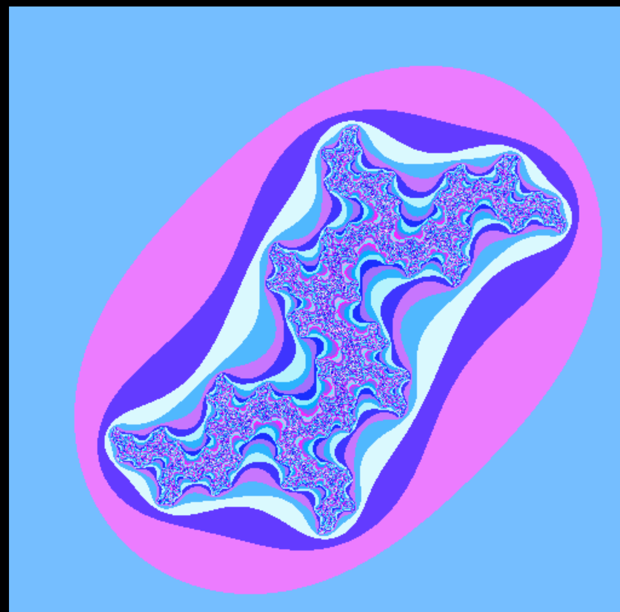




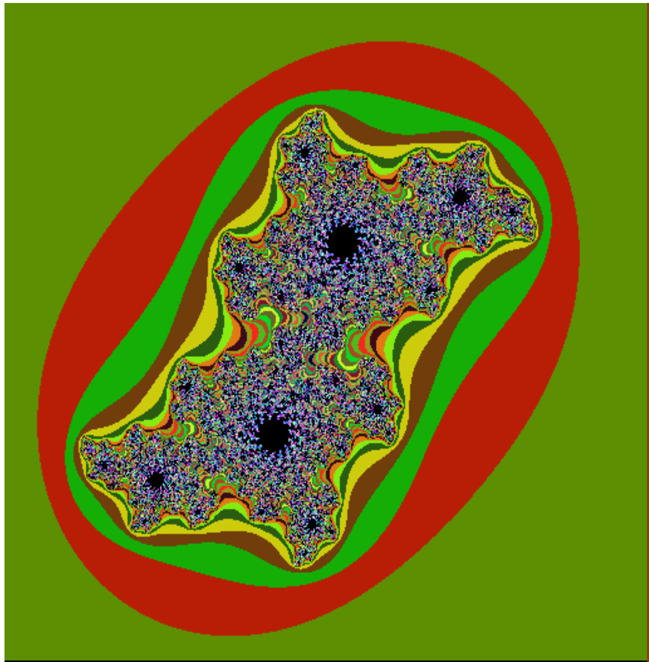


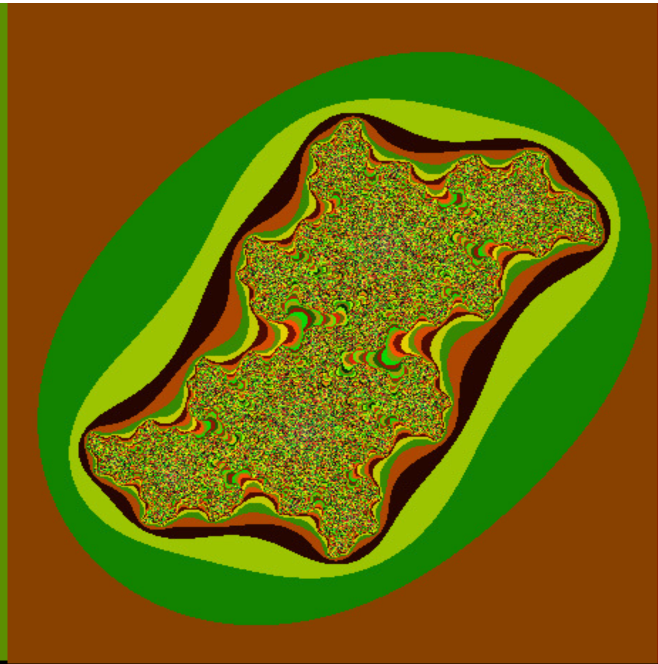
KPG

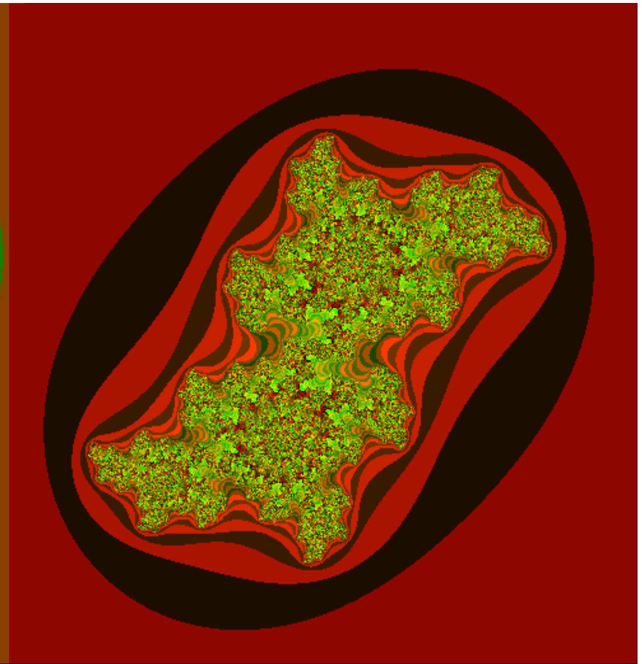




KPG



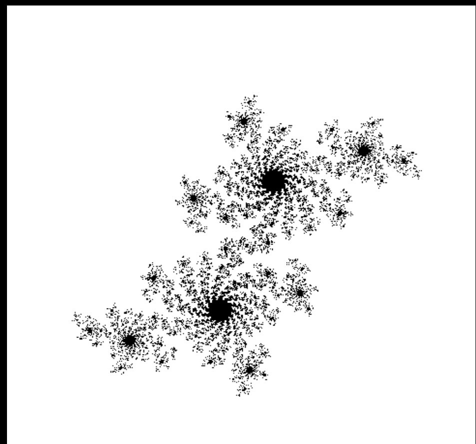




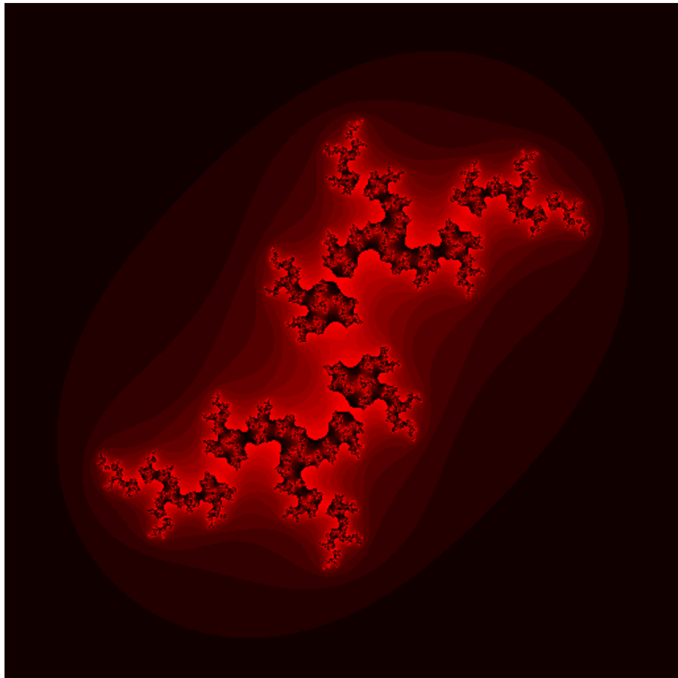
100 iterations, c=0.005+0.65i

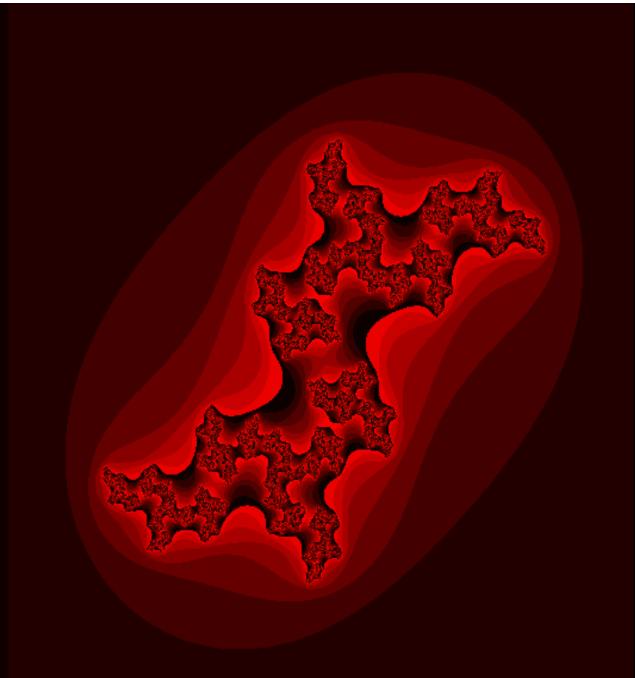
1000 iterations

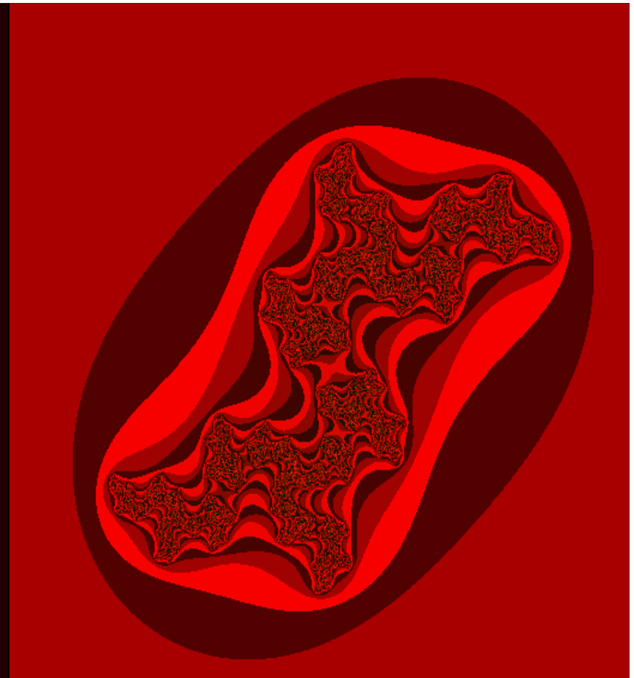
10000 iterations



c=0.005+0.65i



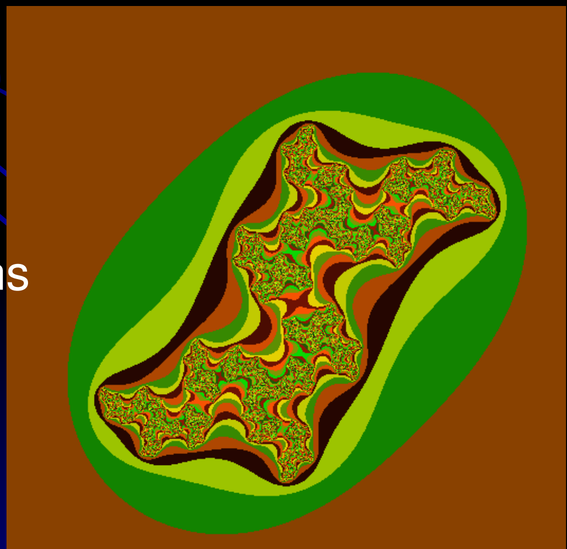




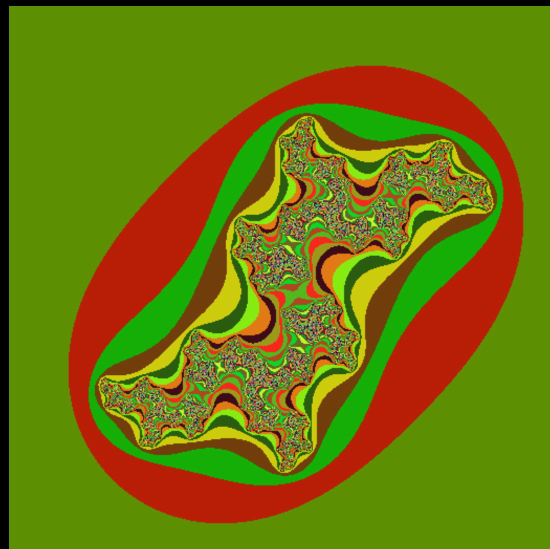
1 mil. iterations

500 000 iterations

100 000 iterations

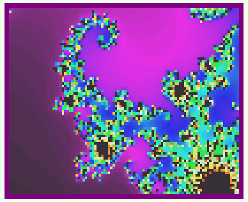


1000 iterations

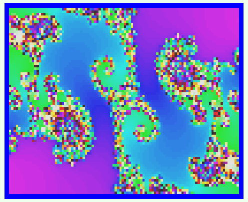


100 iterations

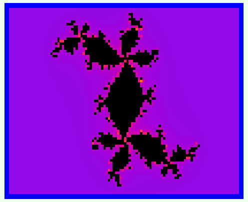
Various equations to inspire



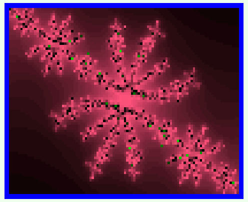
f(z)=z^2 - .74173 + .15518i (195K)



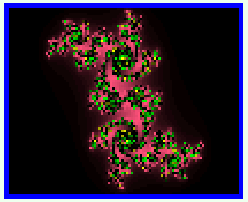
f(z)=z^2 - .74543 + .11301i (484K)



f(z)=z^2 + .29812 + .52923i (35K)

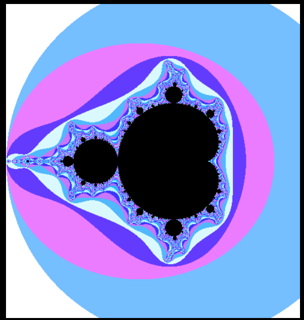


f(z)=z^2 - .55947 + .64196i (105K)



f(z)=z^2 + .23300 + .53780i (104K)

Mandelbrot sets



- B.Mandelbrot, 1980
- We draw the points c=x+iy in the complex plane, for which the function value “remains small"

$$F_{k+1} = F_k^2 + c,\ F_0 = 0 + 0i$$

$$F_1 = c$$

$$F_2 = c^2 + c$$

$$F_3 = (c^2 + c)^2 + c$$

$$F_4 = ((c^2 + c)^2 + c)^2 + c \text{ atd.}$$

- Inspected value: $|F_k|$

- It is inspected whether $|F_k|$ for k=0,1,2,... And given c grows above all limits
- Compute first N iterations, N ~ 1000, often even less
- If $|F_k| \leq 2$ up to Nth iteration, we expect that the point lies in the M. set, we color it black
- If $|F_k| > 2$, the sequence will grow above all limits, the point does not lie in M set, we color it white or according to the number of iterations which $|F_k|$ needed to grow above 2
- The boundary of M. set is a fractal curve
- M. set is continuous

- Ex.: Behaviour for one particular $c = -0.2 + 0.5i$

  $F_2 = 0.41+0.3i$,

  $F_3 = -0.1219+0.254i$

  $F_4 = -0.2497+0.4381$ atd.

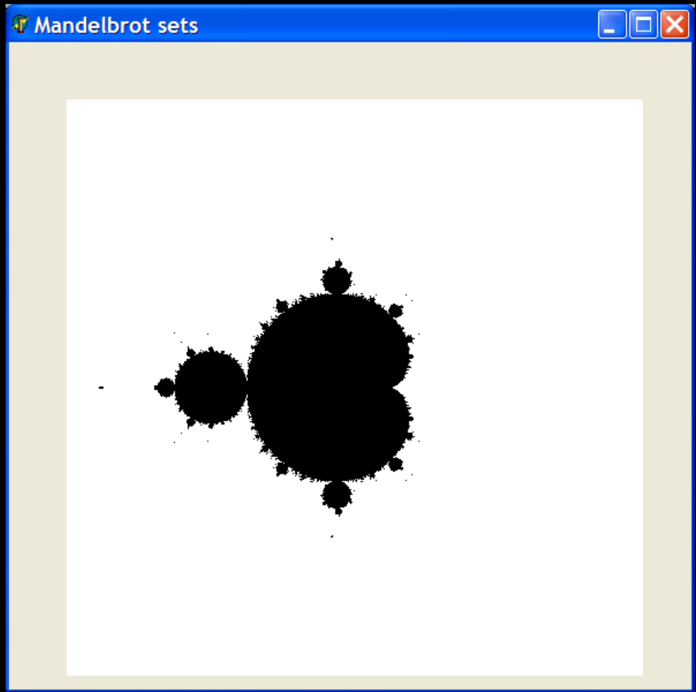
  After ~ 80 iterations it converges to $F_k = -0.2499+0.33368i$

  - fixed point of the function

  $|F_k| = 0.416479$ => c lies in M. set

- For computation use at lest the double type, compute more iterations nearby the boundary
- Although M. set is self-similar in zoom, details are not identical with the whole

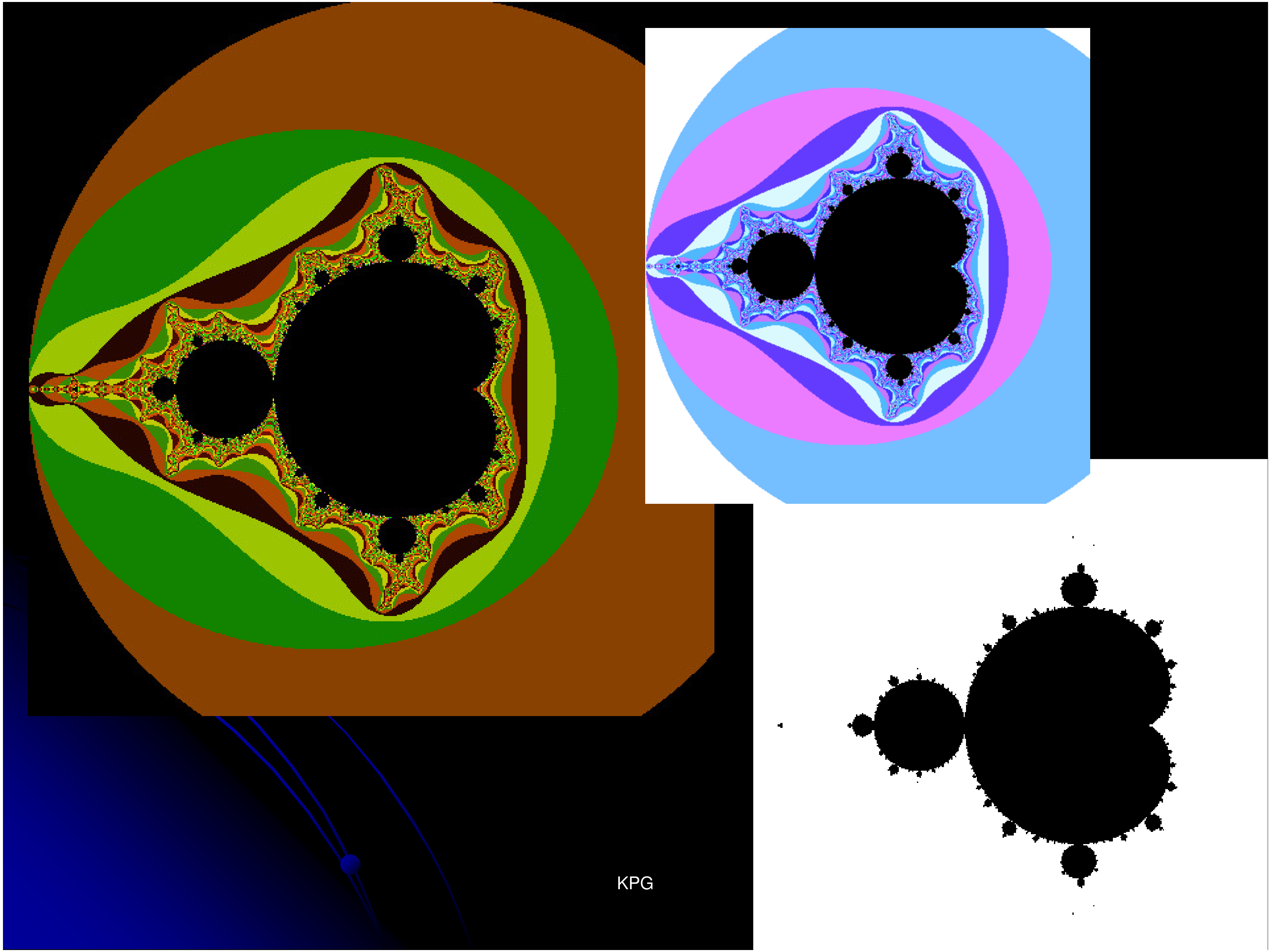
- Definition: Mandelbrot set – a set of all complex c for which the iterations $F(z) \rightarrow z^2+c$ are limited (the start is in $z=0+0i$)
- In other words: a graph of all complex numbers c which are not growing to ∞ when iterated in $F(z) \rightarrow z^2+c$ with the starting value $z=0+0i$.

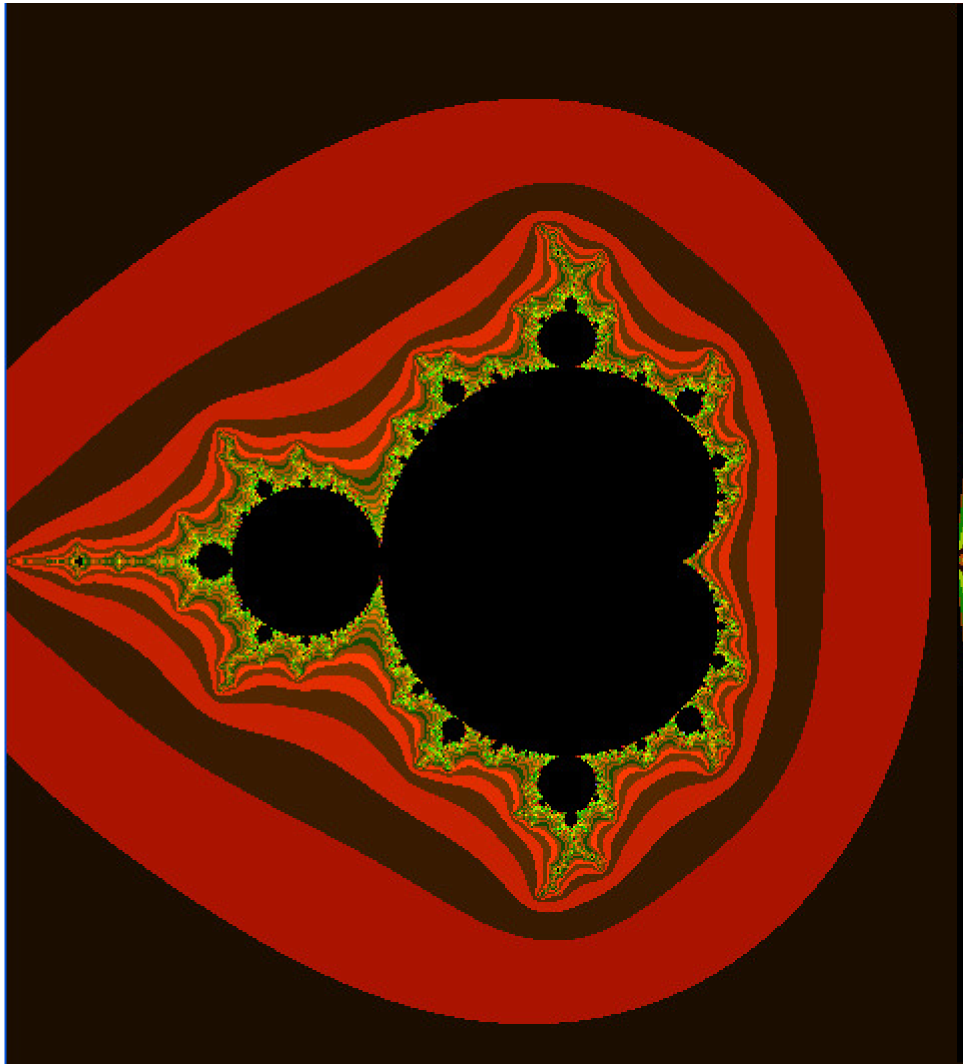


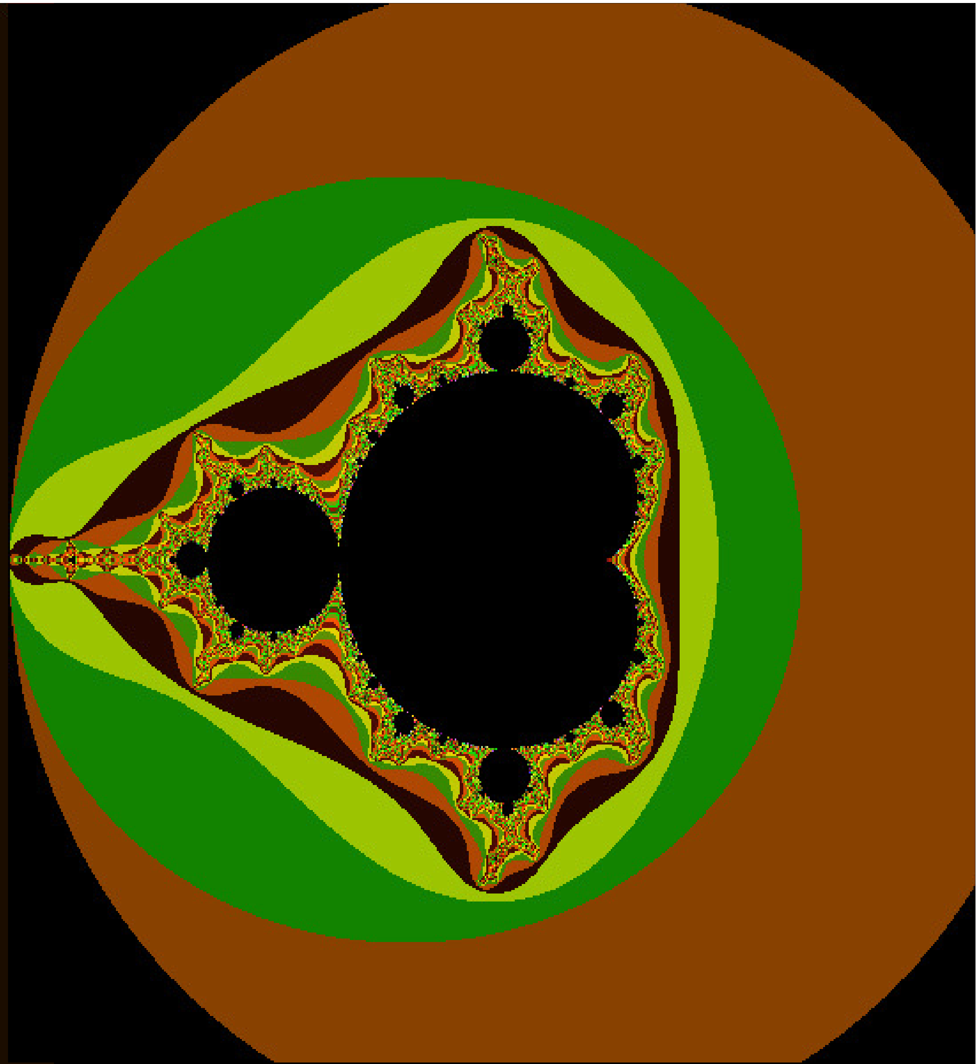
100 iterations
(more iterations would bring more details)

```
function MandelCount (cx,cy: extended; num: longint) : longint;

{ num is the maximum number of iterations }

const thresh = 4.0; { a larger threshold may yield better pictures }

var
   x,y,tmp,fsq : extended;
   count : longint;

begin
 x := cx; y := cy; fsq := x*x+y*y;
 count := 0;
 while (count < num) and (fsq <= thresh) do
       begin
        count := count+1;
        tmp := x;
        x := x*x - y*y + cx;
        y := 2.0*tmp*y + cy;
        fsq := x*x + y*y;
       end;
 MandelCount := count;
end; { MandelCount }
```

```
procedure Fill_pixels (var cells : TArray);

{ procedura spocte Manedlbrotovu mnozinu do pole cells }

var i,j : integer;
    count,num : longint;    cx,cy : real;
    col : TColor;

begin
 nrows := Form1.height; ncols := Form1.width; num := 1000;
 for i := 0 to nrows-1 do
     begin
      cy := ymin+(ymax-ymin)*i/(nrows-1);
      for j := 0 to ncols-1 do
          begin
           cx := xmin+(xmax-xmin)*j/(ncols-1);
           count := MandelCount (cx,cy,num);
           if count=num then cells[j,i] := clBlack { point in the
                                                    Mandelbrot set }
           else
             cells[j,i] := clWhite;
          end;
     end
end; { Fill_pixels }
```


KPG





10 000 iterations

1000 iterations

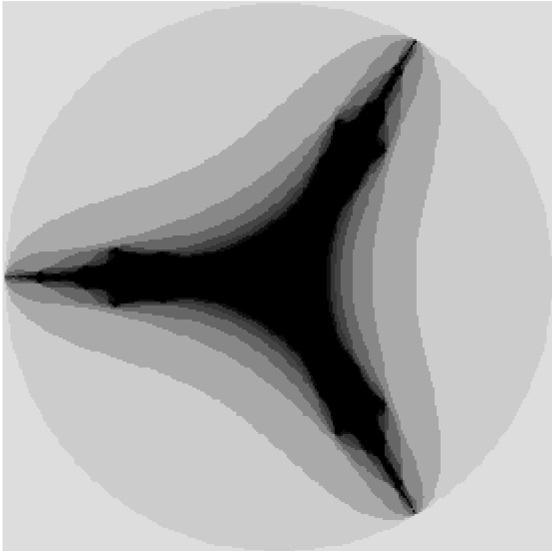
- $z=z^2+c$ – the best known M. set, but other equations also possible – other M. sets
- To get M.set, c musít be a variable and z start in (0,0i)
- Mandelbrot set – various c values in the plane are drawn,

  Juliova set – various starting z values are drawn while c is constant.

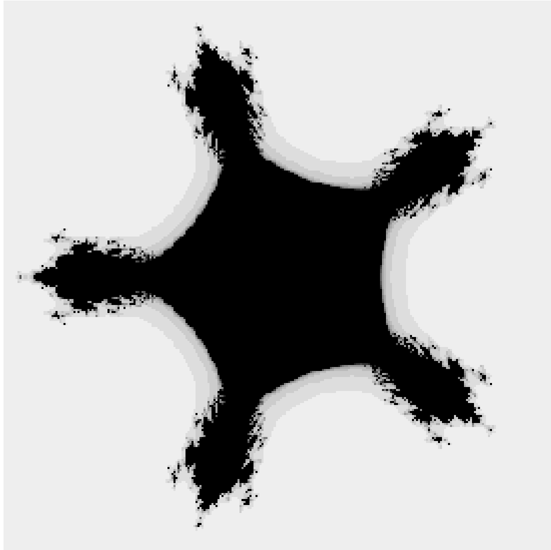
- Use of J. and M. sets: studies of phase transformations, dynamic systems + theory of chaos, biology of evolution

Mandelbar set:

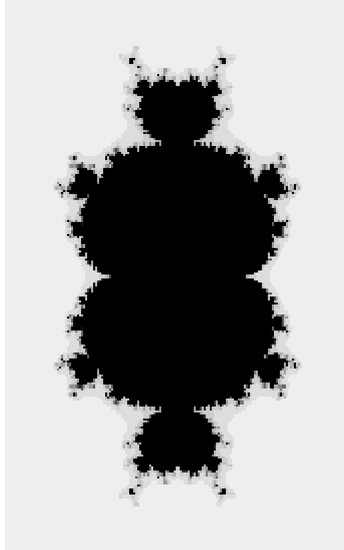
$\overline{M}(2): z \to \overline{z}^2 + c$



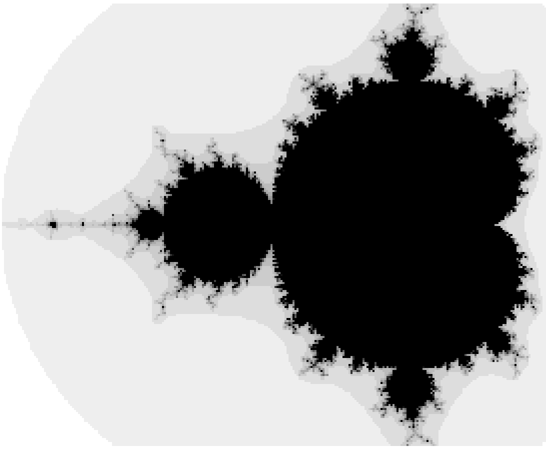
$\overline{M}(4): z \to \overline{z}^4 + c$



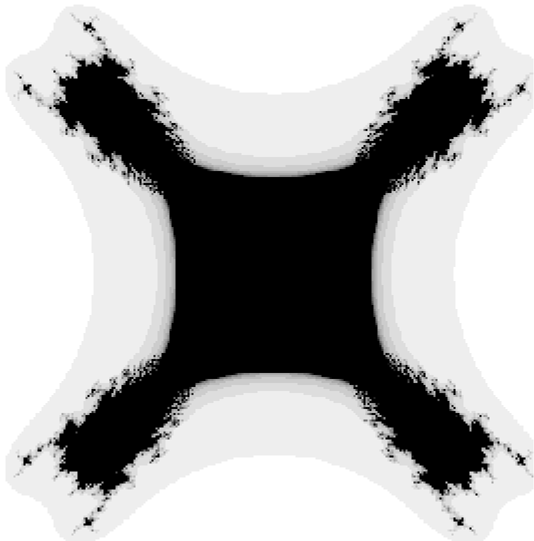
$M(3)$: z --> $z^3 + c$:



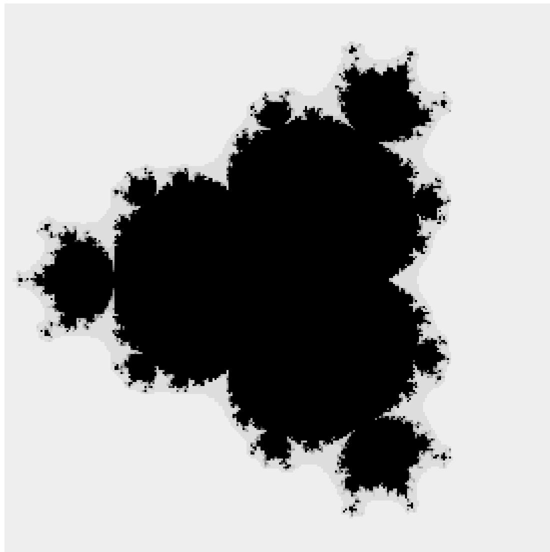
Mandelbrot set ($M(2)$: z --> $z^2 + c$):



$\overline{M}(3): z \to \overline{z}^3 + c$



$M(4)$: z --> $z^4 + c$:



- Modification: authomatic fractals generation – to produce nice pictures (J.C.Sprott, C.A.Pickover, 1995)
  - Take simple equations with randomly chosen coefficients
  - Solve them on a computer
  - Visualize only those which have some “artistic quality"
- General 2D quadratic iterated map:

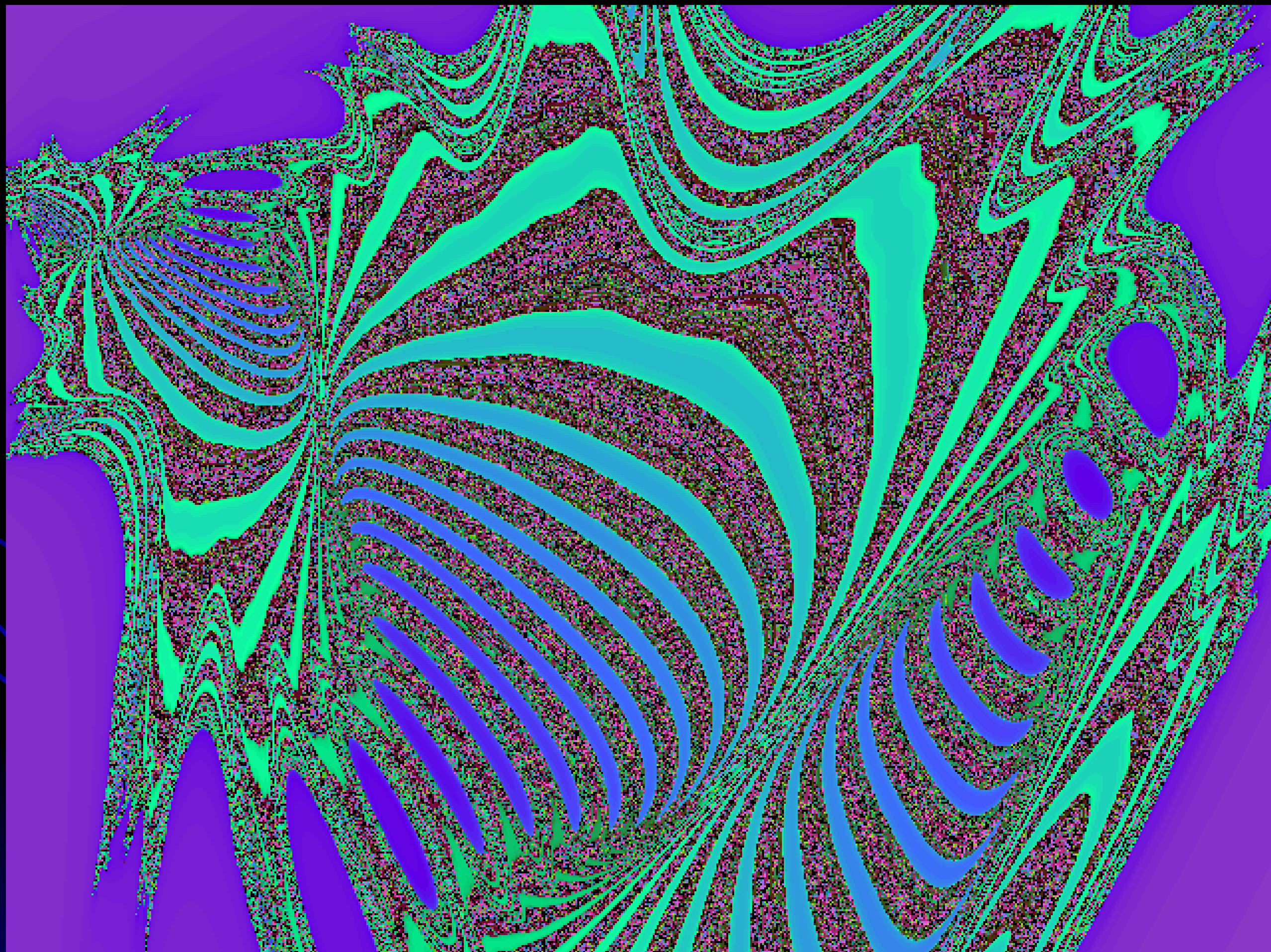
$$x_{new} = a + bx + cx^2 + dxy + ey + fy^2$$

$$y_{new} = g + hx + ix^2 + jxy + ky + ly^2$$

where a-l are randomly chosen, kept constant for computation of one fractal

- Visualization: either to draw (x,y) or solve for various starting values and compute number od iterations needed to leave some area, the color is set according to the number of iterations
- Also possible for Julia sets
- Coefficients: -1.2 až 1.2, inc 0.1the the equations are iterated with the initial condition x=y=0, if within 100 up to 1000 iteration the computation escapes from the circle centred in the origin with r=1000, then we save parameters and compute the so-called Escape fractal for some area
- Time to escape the area can be represented by a height, like a terrain

- Visually interesting fractals: those escaping slowly (100-1000 iterations)



- ~ 1 interesting case of 300