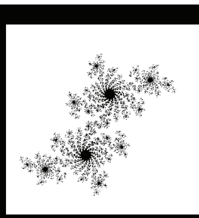
Juliovy a Mandelbrotovy množiny

I.Kolingerová

Zdroje:

- 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

Juliovy množiny

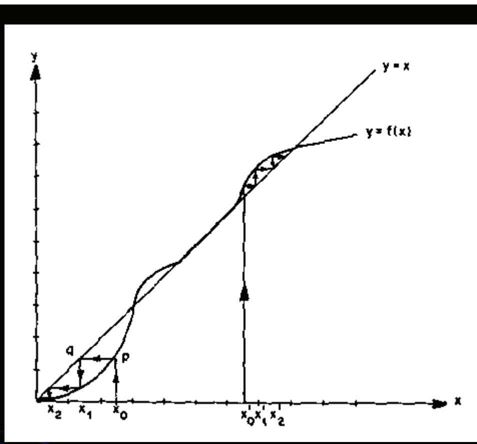


- Francouzský matematik G.Julia, 1918
- J.množina: $z_{n+1} = F(z_n)$, z_n – komplexní číslo
- Zachovává úhly, ale měřítko závisí na hodnotě z (lokálně je to rotace se změnou měřítka, měřítkový faktor $|F'(z)|$
- Standardní příklad: $z_{n+1} = z_n^2 + c$, $c = a+ib$
  v reálné notaci: $x_{n+1} = x_n^2 - y_n^2 + a$,
  $y_{n+1} = 2x_n y_n + b$
- Důležité jsou pevné a periodické body F

KPG 3

- Pevný bod: určen vztahem $z = F(z)$
- Pokud $|F'(z)| < 1$, je bod stabilní. Je-li z_0 blízko stabilnímu pevnému bodu z, pak orbita z0, $z_1, z_2, z_3, \ldots$ konverguje k z. Pak z je přitahující pevný bod – atraktor.
- Pokud $|F'(z)| > 1$, je bod nestabilní, odpuzující.
- Pokud $|F'(z)| = 1$, je pevný bod neutrální.
- Periodický orbit (m-cyklus): $z_m = F(z_{m-1}) = z_0$, m – nejmenší celé číslo, pro které k tomu dojde; z_0 – periodický bod řádu m. M-cyklus je stabilní, pokud $|F'(z_0)F'(z_1)F'(z_2)\ldots F'(z_{m-1})| < 1$, dtto nestabilní.

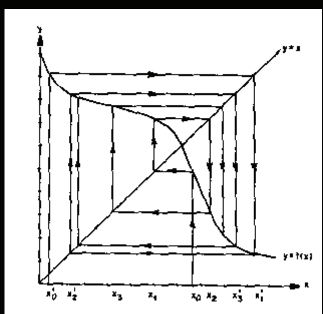
KPG 4



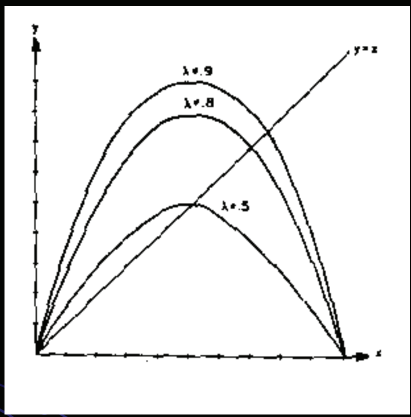
Monotónní funkce, kladná derivace, neprodukuje chaos

Pevné body jsou $x^{(1)} = 0$, $x^{(2)} = 3/10$, $x^{(5)} = 4/5$, interval J=1/2 až 3/5

z poč. bodu x_0, x_0': $x^{(1)}$ a $x^{(5)}$ jsou atraktory, $x^{(2)}$ je repelor, zatímco J přitahuje blízké body vlevo odpuzuje blízké body vpravo,



Monotónní funkce, záporná derivace, neprodukuje chaos



Na fukci f.5 je x=0 repelor a x=0.5 atraktor,

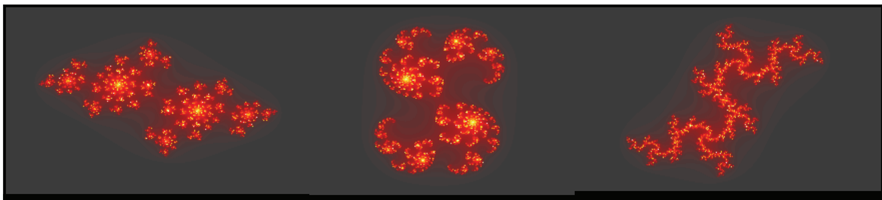
f.8 má x=0 repelor a x=11/16 také repelor a 2-cyklus (x asi 0.51, x asi 0.8) je atraktor

f.9 má periodické orbity 2^n a iterace jsou chaotické

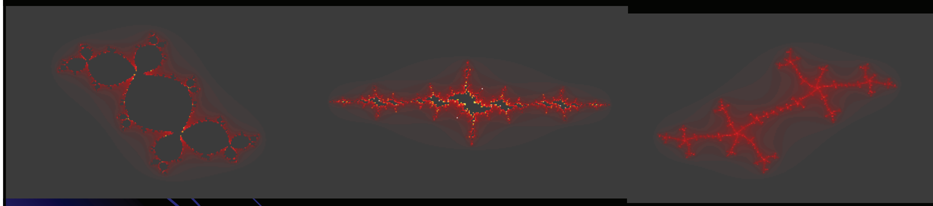
KPG 7

- Definice: Juliova množina – množina všech komplex. čísel z: iterace F(z) ->z^2+c je omezená pro určitou hodnotu c.
- Jednodušeji: graf všech komplex. čísel z, které se neblíží k ∞, když jsou iterovány v F(z) ->z^2+c, kde c je konstanta.
- Vezme se pevné c, určí se barva pixelu v (x,y) použitím z_0=x+iy jako počát. bod
- Více iterací – větší detail v obrázku
- Vzhled podle toho, zda c leží na odpovídající Mandelbr. množině – pokud ano, J. mn. bude zcela souvislá, a naopak (tj. rozhoduje orbita pro c=0).

KPG 8



Juliovy množiny pro $F(z) \to z^2+c$



KPG 9

- Výpočet:

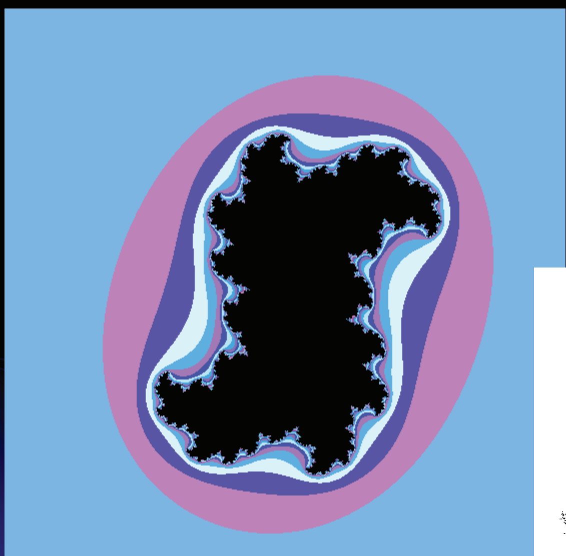
```
function JuliaCount (x,y: extended; num: longint) : longint;

{ 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 }
```

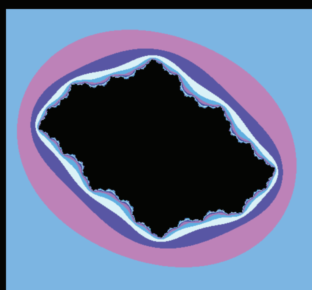
KPG 10

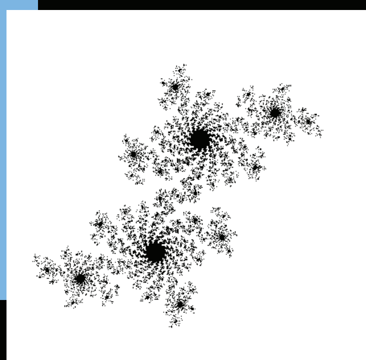
- Celek:

```
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 }
```

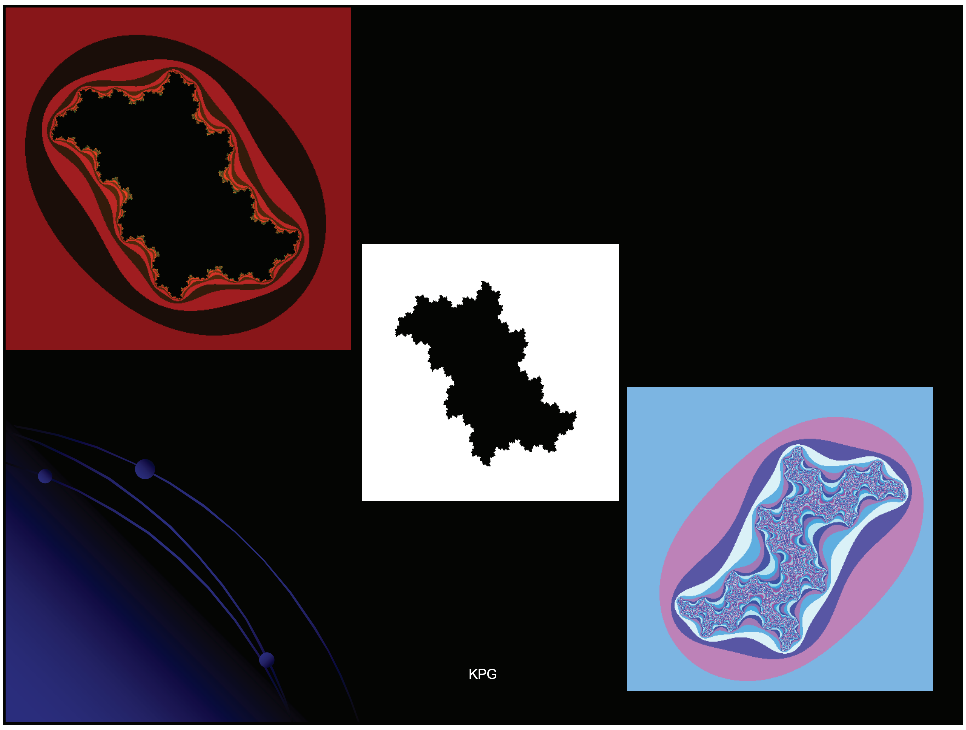
KPG 11

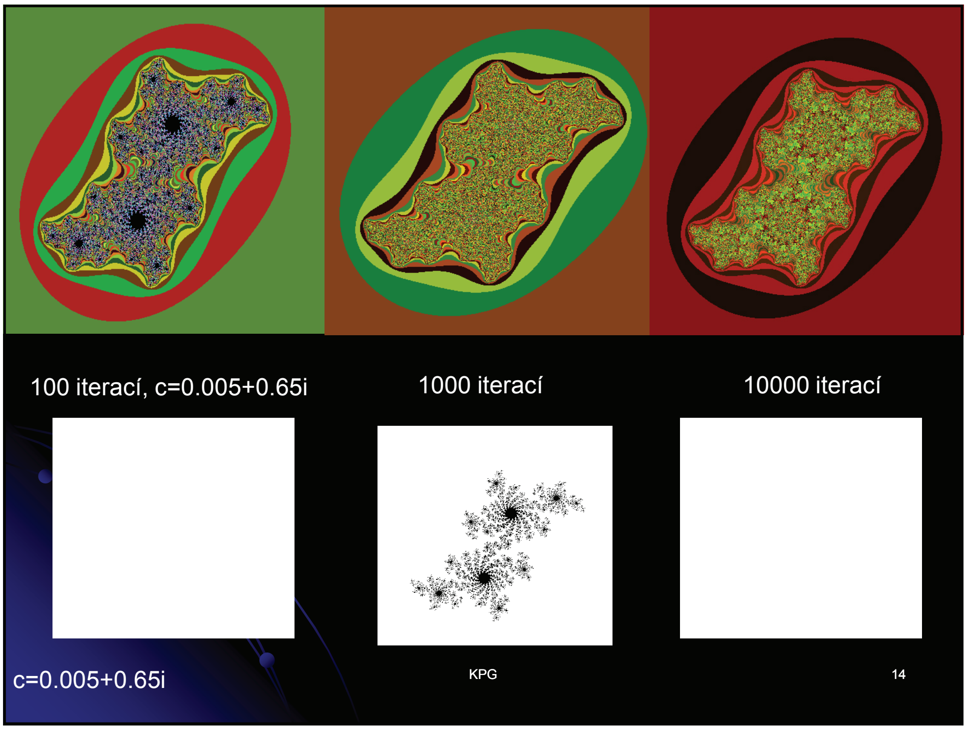


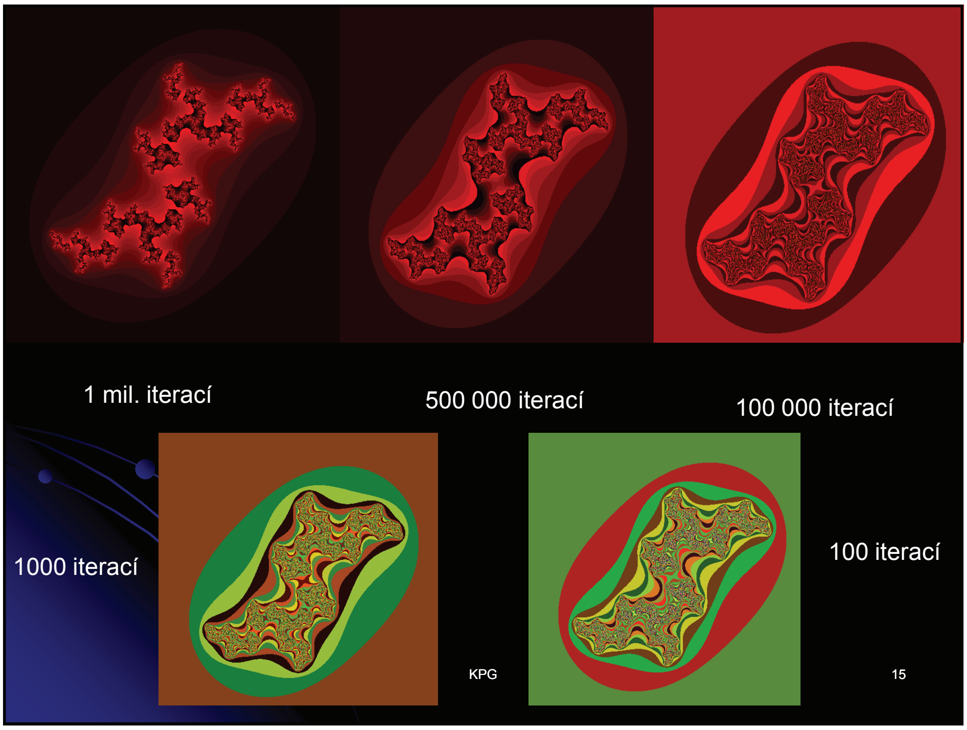




KPG

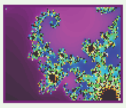

KPG

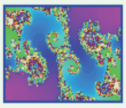

100 iterací, c=0.005+0.65i
1000 iterací
10000 iterací
c=0.005+0.65i
KPG
14



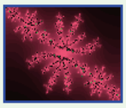
Různé rovnice pro inspiraci

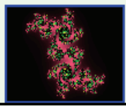
(http://www.jcu.edu/math/faculty/spitz/juliaset/juliaset.htm)


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

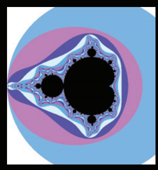

f(z)=z^2 - .74543 + .11301i (484K)


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


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


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

Mandelbrotovy množiny



- B.Mandelbrot, 1980
- Kreslí se body c=x+iy v komplex. rovině, pro které hodnota funkce "zůstává malá"

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

  $F_1 = c$

  $F_2 = c^2 + c$

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

  $F_4 = ((c^2 + c)^2 + c)^2 + c$ atd.
- zkoumaná hodnota: $|F_k|$

KPG 17

- Zkoumá se, zda $|F_k|$ pro k=0,1,2,... a dané c roste nade všechny meze
- Spočítat prvních N iterací, N cca 1000, často stačí méně
- Pokud $|F_k| \leq 2$ do N=té iterace, předp. se, že bod leží v M. množině, obarvíme černě
- Pokud $|F_k| > 2$, posloupnost poroste nade všechny meze, bod neleží v M. množině, obarvíme bíle nebo podle počtu iterací, kt. $|F_k|$ potřebovalo, než překročilo 2
- Hranice M. množiny je fraktální křivka
- M. množina je souvislá

KPG 18

- Př.: Chování pro 1 konkrétní c = -0.2 + 0.5i

  F_2 = 0.41+0.3i,

  F_3 = -0.1219+0.254i

  F_4 = -0.2497+0.4381 atd.

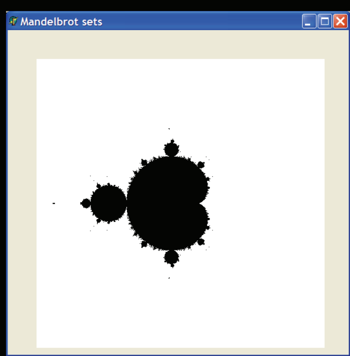
  Po asi 80 iteracích konerguje k F_k = -0.2499+0.33368i

  - pevný bod funkce

  $|F_k|$ = 0.416479 => c leží v M. množině

- Formát pro výpočet: aspoň double, blízko hranice více iterací
- Ačkoliv je M. množina při zvětšení soběpodobná, detaily nejsou identické s celkem

- Definice: Mandelbrotova množina – množina všech komplex. čísel c: iterace F(z) ->z^2+c je omezená (počátek v z=0+0i)
- Jednodušeji: graf všech komplex. čísel c, které se neblíží k ∞, když jsou iterovány v F(z) ->z^2+c s poč. hodnotou z=0+0i.



100 iterací

- Výpočet:

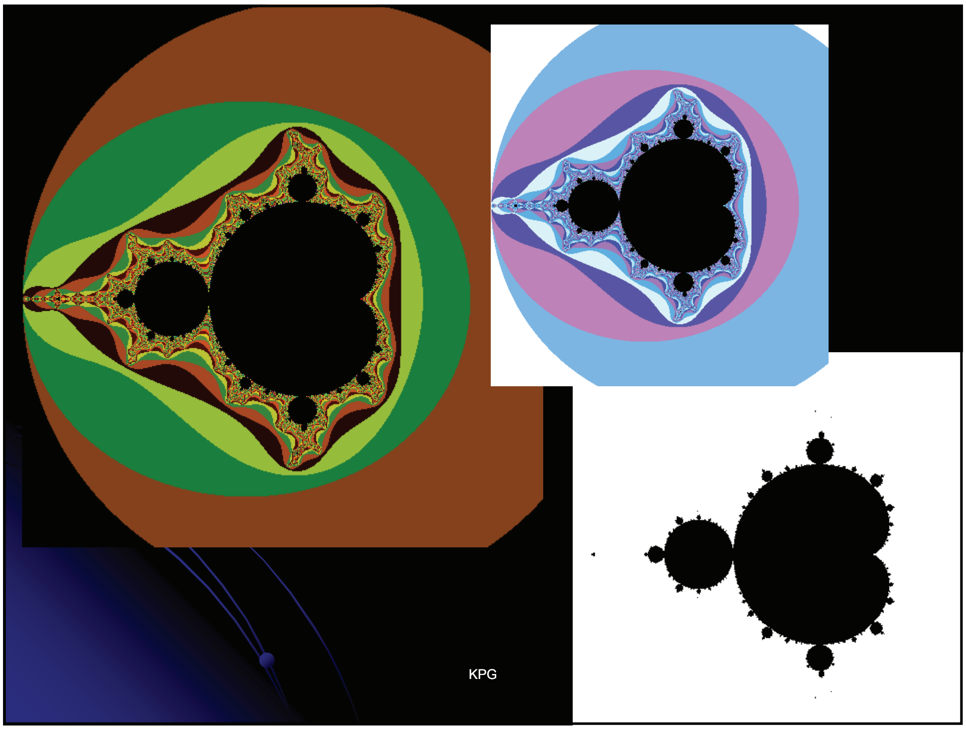
```
function MandelCount (cx,cy: extended; num: longint) : longint;

{ 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 }
```

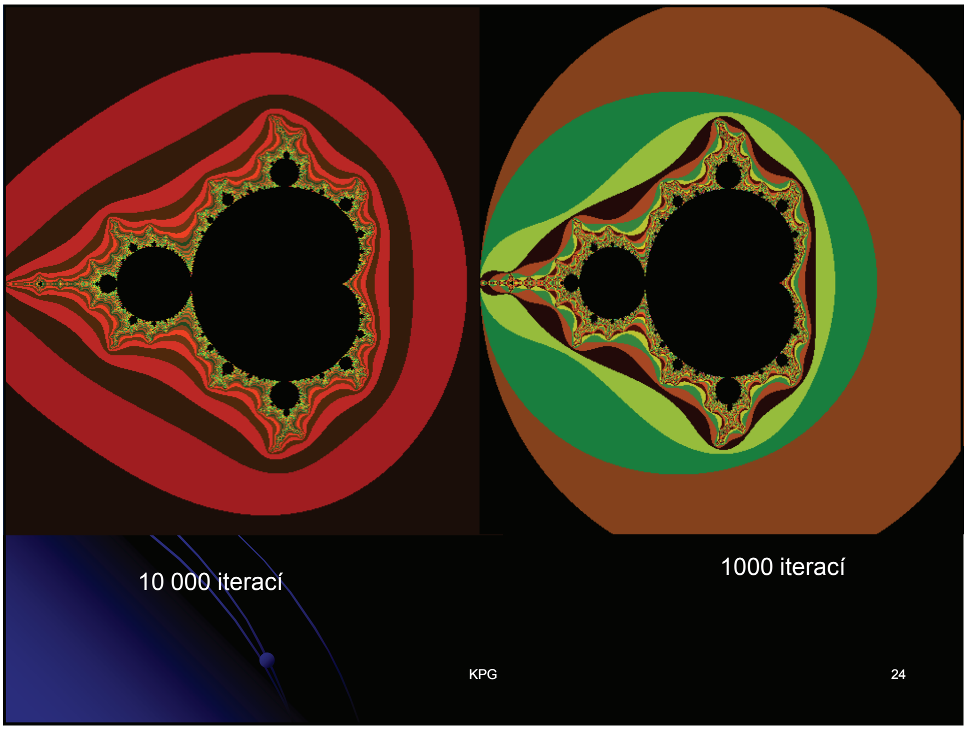
KPG 21

- Celek:

```
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 }
```

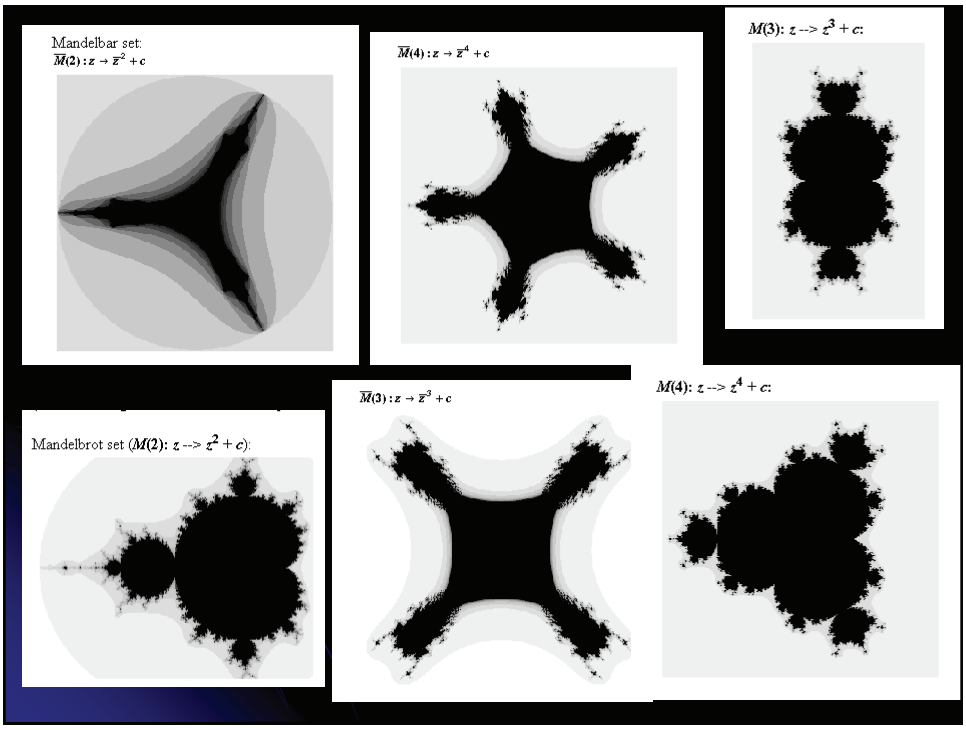
KPG 22


KPG


10 000 iterací
1000 iterací
KPG
24

- $z=z^2+c$ – nejznámější M. množina, ale jiné vzorce také možné – jiné M. množiny
- Aby to byla M.mn., c musí být proměnná a z začínat v (0,0i)
- Mandelbrotova množina – kreslí se různá c v kartézské rovině, Juliova množina – kreslí se různé počáteční hodnoty z, přičemž c je konst.
- Užití J. a M. množin: studie fázových transformací, dynamické systémy + teorie chaosu, vývojová biologie

KPG 25



- Modifikace: automatická generace fraktálů pro umělecké účely (J.C.Sprott, C.A.Pickover, 1995)
  - vzít jednoduché rovnice s upravitelnými náhodně vybranými koeficienty
  - vyřešit je na počítači
  - zobrazit jen ty, které splňují kritéria indikující "uměleckou kvalitu"
- obecná 2D kvadratická iterovaná mapa:

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

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

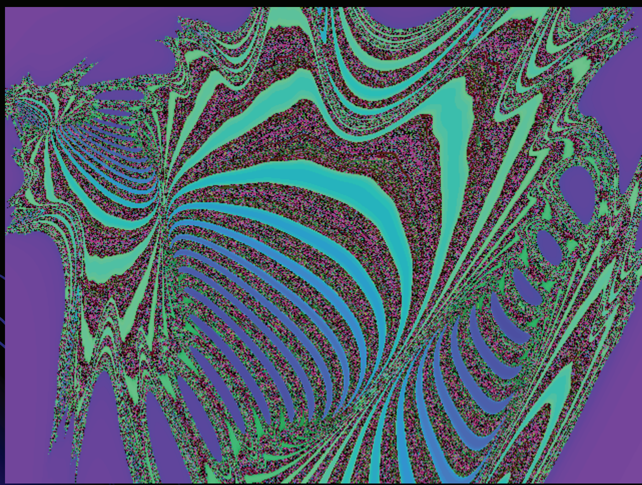
kde a-l náhodně volené konst., během výpočtu 1 fraktálu beze změny

KPG 27

- Zobrazení: buď kreslit (x,y) nebo řešit pro různé počát. hodnoty a počítat počty iterací potřebných pro opuštění nějaké oblasti, barvu nastavit podle počtu iterací
- Též pro Juliovy množiny
- Volba koef: -1.2 až 1.2, inc 0.1, pak iterace rovnic s poč. podm. x=y=0, pokud opuštění kružnice se středem v počátku, r=1000, za100 až 1000 iterací, pak uložíme parametry a spočteme Escape fraktál pro určitou oblast
- Parametry reprezentovat písmeny (A=-1.2 atd.)
- Čas úniku možno reprezentovat výškou (terén)

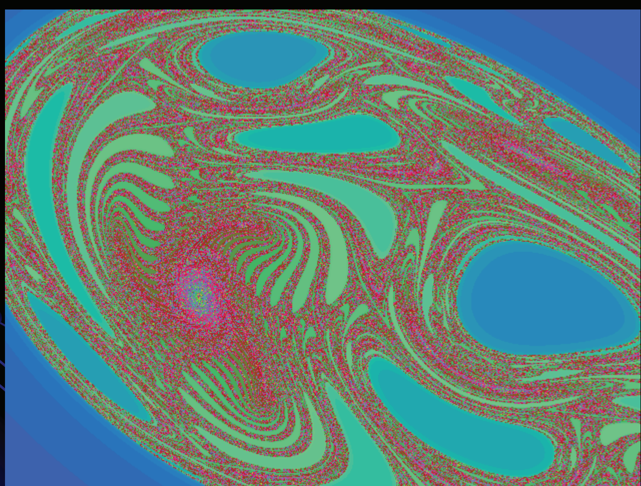
KPG 28

- Vizuálně zajímavé fraktály: ty, pro něž orbity oblast opouštějí pomalu (100-1000 iterací)



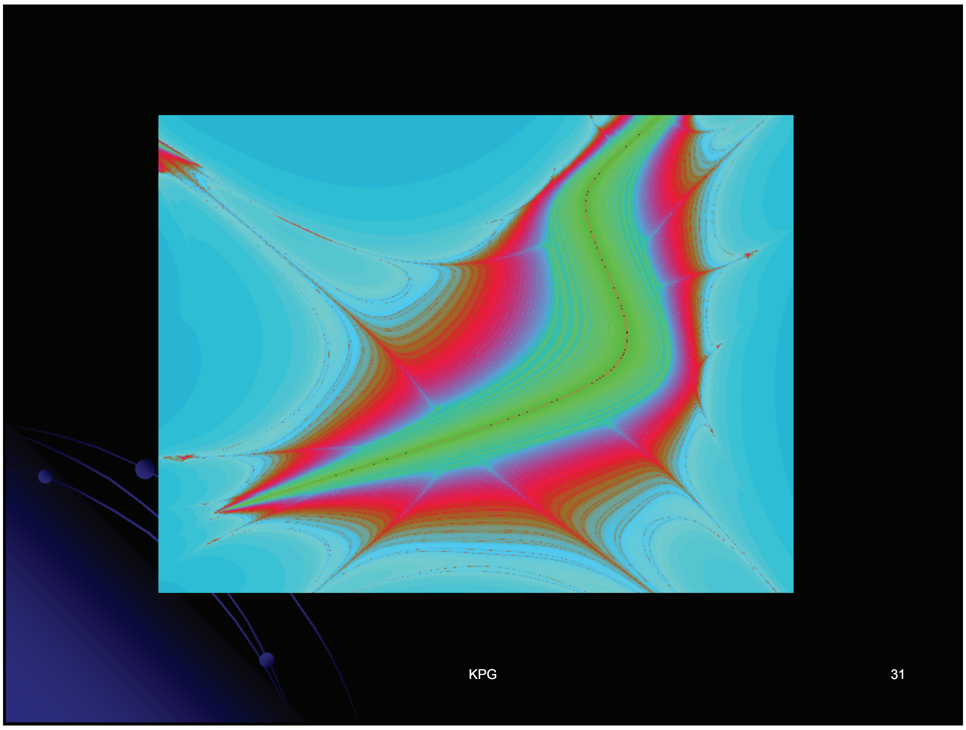
29

- cca 1 zajímavý případ z 300



KPG

30


KPG
31