

10. FRAKTÁLY



Cíl Po prostudování této kapitoly budete znát

- principy fraktální grafiky na osobních počítačích
- použití fraktálů v počítačové grafice
- algoritmy tvorby fraktálů



Výklad

Dosavadní dělení geometrie dle rozměru - dvoj, trojrozměrného je dělení podle eukleidovské podmnožiny tohoto prostoru. Při vícerozměrné geometrii se hovoří o tzv. topologické dimenzi. Dimenzi definujeme tak, že vícedimenzionální útvar lze rozdělit útvarem o dimenzi nižším na dva disjunktní útvary. Na příklad. Krychli - třídimenzionální útvar - lze rozdělit rovinou - dvoudimenzionálním útvarem - na dvě části. A podobně. Topologie se zabývá tím, co se při spojitých transformacích nemění. Naše dříve uvedené transformace - posunutí, měřítko, otočení a pod. jsou vesměs zvláštní případy afinních transformací. V přírodě se však tyto pravidelné útvary - čtverce, kružnice, a pod. , tak i "naše" transformace nevyskytují. Spíše se v přírodě vyskytují nepravidelné vlastnosti, které se topologicky velmi obtížně popisují. Objekty zde jsou hory, mraky, stromy,...., pohyb - např. Brownův pohyb mikroskopických částic. Tedy útvary náhodné - nepodobné; pohyb chaotický. Otázky, které si klade fraktální geometrie jsou: je opravdu tento pohyb pouze chaotický a útvary nepodobné?

10.1. Soběpodobnost

V šedesátých letech byl zaveden pojem **soběpodobnosti**. Mírou soběpodobnosti byla zavedena **Hausdorffova dimense**. Tedy teorie **soběpodobnosti** se stala základem **fraktální geometrie**, která se zabývá generováním soběpodobných útvarů. Vlastní pojem fraktál (fractus = polámaný, nepravidelný) zavedl v osmdesátých letech matematik polského původu **Benoit B. Mandelbrot**, kterému se podařilo nalézt souvislost mezi soběpodobnými útvary v matematice a přírodními útvary. Ukázal, že co v matematice do té doby bylo považováno za domýšlení myšlenkových konstrukcí a absurdnostem, může se stát základem pro modelování přírodních útvarů.

Příklad soběpodobných útvarů - objektů: přírodní kámen. Vezmeme-li dva kameny - velký a malý, potom tyto se od sebe liší právě velikostí. Matematicky řečeno změnou měřítka. Nebo na níže uvedeném obrázku je znázorněna oblast - sněhová vločka - ohraničení plochy křivkou, která vznikne geometrickou konstrukcí dle pravidel geometrie. Prostředky počítačové grafiky umožnily matematice vlastně zcela rozvinout fraktální geometrii. Pomocí počítače je možno matematickou teorii prakticky reálně ověřit a zkoumat. Naopak počítačová grafika dostala další aparát, pomocí něhož lze generovat a zaznamenat obrazy, které by jen ztěžší mohly jinak vzniknout.

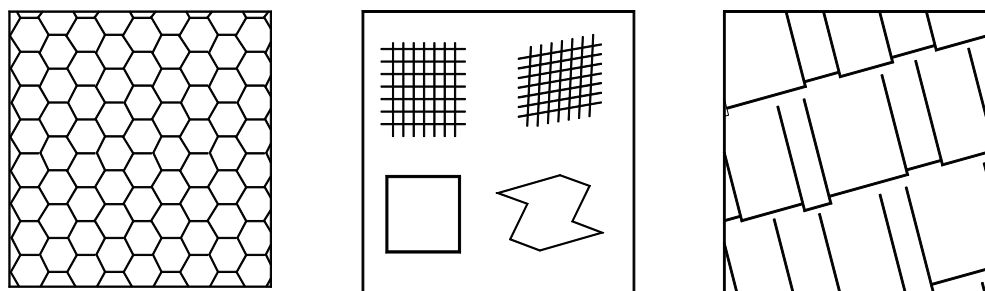
Matematicky je soběpodobná množina definována:

Soběpodobná množina A n -dimenzionálního Eukleidovského prostoru E je taková množina, pro níž existuje konečně mnoho kontrahujících zobrazení $\varphi_1, \dots, \varphi_m$ takových, že A vznikne jako sjednocení*

$$A = \bigcup_{i=1}^m \varphi_i(A).$$

Soběpodobná množina vznikne tedy opakováním sebe sama při určité transformaci. To je na příklad posunutí, změnu měřítka, rotace a pod. Soběpodobné množiny jsou invariantní vzhledem k těmto transformacím. To znamená, že změnou měřítka jsou množiny stejné. Zvětšením ani zmenšením se množina nemění. To potom znamená, že množiny jsou soběpodobné. Mimo jiné to znamená, že množina, která vznikla opakováním stejného motivu je **soběpodobná**.

* kontrahující zobrazení (Rektorys a kol.: Přehled užití matematiky).



Obr. 10.1

Příklady soběpodobných množin jsou na následujících obrázcích (Obr.10.1).

Ve fraktální geometrii se používá pojem **topologická** dimenze. Jde o dimenzi, kterou chápeme tak, že spojitým zobrazením každé křivce přiřadíme přímku; ploše rovinu a pod.

Hausdorffova dimenze potom je mírou **členitosti**. Čím je množina členitější, tím je vyšší její Hausdorffova dimenze.

Příklady fraktálů.

10.2. Cantorovo diskontinuum

V roce 1883 Georg Cantor popsal množinu, jejíž algoritmus by se dal vyjádřit:

program Cantor;

procedure rekurzivně_rozděl (X: úsečka);

begin

 rozděl X na A, B, C - tři stejně dlouhé úsečky;

 rekurzivně_rozděl (A);

 rekurzivně_rozděl (C);

end; {rekurzivně_rozděl }

begin

 vezmi_úsečku X;

 rekurzivně_rozděl (X)

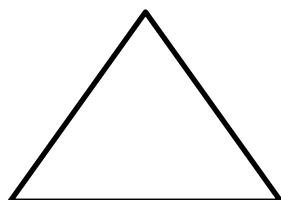
end.

První čtyři iterace Cantorova diskontinua jsou na následujícím obrázku (Obr. 10.2).



Obr. 10.2

10.3. Sněhová vločka - oblast ohraničená křivkami Kochové. (Obr. 10.3)



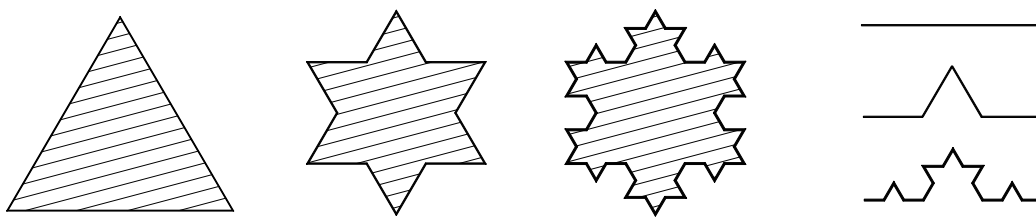
počáteční stav



generátor

Obr. 10.3

Zobrazení počátečních kroků.



Obr. 10.4.

Konstrukce Kochové křivky je

- rozděl úsečku na n (3) díly;
- nad středním dílem sestroj dvě strany rovnostranného trojúhelníka.

Rekurzí můžeme tuto konstrukci vytvářet do určité hloubky. Pokud do procesu nezahrneme náhodný proces, vznikne pravidelný obrazec.

program vlocka_Koch;

procedure rekurzivně_iteruj(X);

begin

rozděl X na tři třetiny A, B, C;

na střední částí sestroj rovnostranný trojúhelník se stranami B1 a B2;

rekurzivně_iteruj(A);

rekurzivně_iteruj(B1);

rekurzivně_iteruj(B2);

rekurzivně_iteruj(C);

end;

begin

vezmi úsečku X;

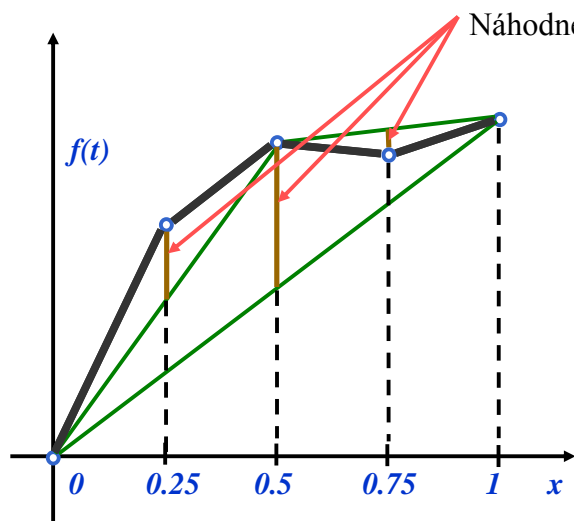
rekurzivně_iteruj (X)

end.

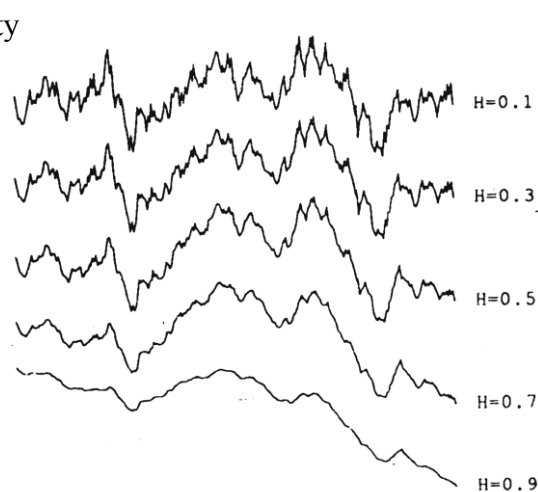
10.3. Brownův pohyb

Začleněním náhodného procesu, ať se jedná o hodnotu nebo úsek programu vytvoří útvar, který je "přirozenější". Celý tento algoritmus nazýváme **transformační funkcí**.

Příklad **náhodného - přírodního** - je možno ukázat na matematických modelech Brownova pohybu (pohyb mikroskopických částic v tekutině). Na obrázku 2 je zobrazena křivka



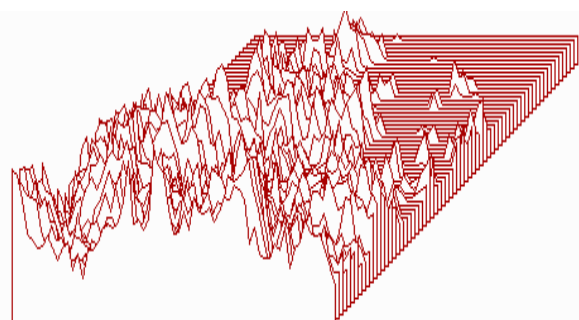
Obr. 10.5



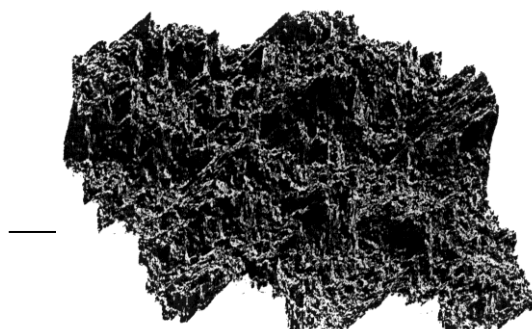
Obr. 10.6

Brownova pohybu po druhém iteračním kroku pro tento pohyb v jedné dimenzi. Algoritmus tohoto pohybu je založen na náhodném nahrazení středního bodu. Označíme $f(t)$ polohu bodu v závislosti na parametru t , $0 < t < 1$. Nechť $f(0) = f_0$. Hodnotu $f(1)$ zvolíme pomocí generátoru náhodných čísel s normálním rozložením se střední hodnotou 0 a s rozptylem σ^2 . Tím jsme realizovali první krok algoritmu a pohyb částice jsme aproximovali úsečkou $f(0), f(1)$. Dalším krokem aproximujeme pohyb lomenou čarou tak, že vychylujeme střední bod úsečky, ale již s menším rozptylem $0.25 \sigma^2$, obecně potom v k -tém kroku s rozptylem $(0.5)^{(n+1)} \cdot \sigma^2$.

Takto vygenerované fraktální křivky mají Hausdorffovu dimenzi 1.5. Upravíme-li algoritmus tak, že rozptyl v každém kroku $(0.5)^{2H} \cdot \sigma^2$ pro $0 < H < 1$ na místo původního



Obr. 10.7



Obr. 10.8

(0.5)^(n + 1) σ^2 . Budeme-li modifikovat nejen střední hodnoty, ale všechny ostatní, dostaneme fraktální křivky dimenzí 2-H (Viz obr.6).

Na obrázku 10.7 je zobrazen "Model" krajiny vytvořený metodou náhodného nahrazení středního bodu čtvercové sítě.

10.4. L-systémy

Odvozeno od anglického LOGO-like turtle. Což byla hra s kybernetickou želvou. Tato uměla interpretovat několik základních příkazů.

Želva - stroj, který bude umět:

- pohybovat se dopředu;
- otočit se o úhel - doprava - doleva;
- uložit stav do zásobníku;
- vyzvednout stav ze zásobníku.

Stavem želvy budeme tedy rozumět - její polohu (bod na obrazovce) a - směr otočení ...

Přepisovací mechanismus

Neterminální symbolem - proměnnou - což bude znak z nějaké abecedy N, na něž lze aplikovat **pravidlo**.

Terminální symbolem bude znak z abecedy T, na který **není** možno aplikovat pravidlo. Jde tedy o koncový - nepřepisatelný - znak operace přepsání.

Startovní symbol je neterminální symbol, od kterého začíná přepisování.

Pravidlo - zobrazení

$$S \Rightarrow \alpha,$$

kde S je z množiny (T) a α je posloupnost terminálních a neterminálních symbolů.

Přepsáním budeme rozumět aplikaci **pravidla** tak dlouho, dokud je to možné. To je pokud řetězec obsahuje neterminální symboly.

L-systémem tedy rozumíme mechanismus, který aplikuje pravidla.

Příklad 1. množina terminálních symbolů: $T = \{a,b,c\}$

množina neterminálních symbolů: $N = \{A,B\}$

startovací symbol: $A \in N$

množina pravidel:

1. $A \Rightarrow aAb,$
2. $A \Rightarrow B,$

3. $B \Rightarrow cB$,

4. $B \Rightarrow c$

umožní generovat řetězce: $aaacbbb$ (11124)
 $accccb(123334)$ a pod.

Tyto řetězce jsou vygenerovány prepisováním startovního symbolu dle pravidel uvedených v závorce.

Realizace L-systému.

Mějme množinu $T = \{ +, -, f, (,) \}$ a interpretaci jednotlivých znaků danou tabulkou

symbol	instrukce
f	pohyb dopředu
+	otočení doprava o D
--	otočení doleva o D
(uložení stavu do zásobníku
)	vybrání stavu ze zásobníku.

Příklad 2.

množina terminálních symbolů: $T = \{F\}$
 množina neterminálních symbolů: $N = \{f, +, -\}$
 množina pravidel:

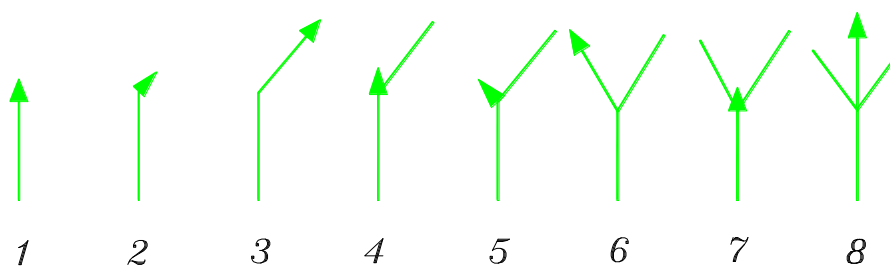
1. $F \Rightarrow F + F - - F + F$
2. $F \Rightarrow f$,

pro úhel otočení $\pi/3$ a startovací řetězec generují tato pravidla sněhovou vločku.

Příklad 3. (se závorkami)

Pohyb podle řetězce $f(+f)(-f)f$ bude:

1. O krok dopředu
2. Uložení stavu na zásobník a otočení doprava
3. Pohyb dopředu
4. Vyzvednutí stavu - návrat do bodu 2.
5. Uložení stavu na zásobník a otočení doleva
6. Pohyb o krok vpřed
7. Vyzvednutí stavu - návrat do bodu 5.
8. Krok dopředu - rovně.



Obr. 10.9

Konkrétní aplikace řetězce $F \Rightarrow F(+F)(-F)F$ (stromek) je v ukázkovém programu L_SYSTEM.PAS.

10.5. Systém iterovaných funkcí IFS - systém

Metoda byla publikována v roce 1985 (Demko) a v roce 1987 (Barnsley). Nejdříve byla tato metoda používána pro generování textur. Nicméně později se stala základem pro metody komprese dat. Základem metody je interaktivním procesem kresba bodů na obrazovce. Interakce spočívá ve výpočtu polohy „nového“ bodu na základě transformace bodu předcházejícího dle transformačního předpisu.

Matematický zápis můžeme formalizovat:

Je dána afinní transformace w v \mathbb{R}^2 , $w: \mathbb{R} \rightarrow \mathbb{R}$, která je popsána maticovým zápisem:

$$w(X) = AX + B$$

čili

$$w\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

rotace, zkosení, měřítko posun

Při Eukleidovské metrice $\rho(x) = \sqrt{x_1^2 + x_2^2}$, $X = (x_1, x_2)$, potom každé afinní transformaci w můžeme přiřadit jediné reálné číslo s následovně:

$$\rho(w(X) - w(Y)) < s \cdot \rho(X - Y),$$

kde s je nejmenší možné reálné číslo, které splňuje výše uvedenou podmínku. Velikost koeficientu s je důležitá pro typ transformace.

Pro $s < 1$ je w kontrakcí

$s = 1$ je w symetrií

$s > 1$ je w extrakcí.

Druhá skupina algoritmů generování fraktálních objektů vychází z modelů dynamických systémů, t.j. systémů, jejichž stav se změní v čase. Mohou to být modely různých biologických, chemických nebo fyzikálních dějů (genetický model populací, průběh koncentrace látek, pohyb planet a pod.). Průběh mnohých algoritmů může mít také dynamický charakter. Buď se ustálí po určitém čase na nějakém "konečném" stavu - hodnotě. Nebo je nestabilní a jeho hodnoty divergují nade všechny meze. Z hlediska fraktální geometrie jsou zajímavé takové systémy, kdy se stavové veličiny mění chaoticky a i malá změna počátečních podmínek vede k zcela rozdílným hodnotám.

Budeme-li se zabývat pouze diskretními dynamickými systémy, potom tento diskretní systém je určen množinou stavů X a zobrazením $f: X \rightarrow X$, $x \in X$. Zobrazení f představuje přechod ze stavu x do stavu $f(x)$ v následujícím časovém okamžiku. Nás bude zajímat chování systému v závislosti na počátečním stavu x_0 . Budeme se tedy zajímat, jaké jsou posloupnosti stavů

$$x_0, x_1 = f(x_0), x_2 = f(x_1) = f(f(x_0)) \text{ v závislosti na } x_0.$$

V teorii dynamických systémů se zavádí pojem **atraktor**. Je to množina stavů, k níž systém za určitých okolních stavů vždy konverguje. Pojem **podivný atraktor** se zavádí tehdy, jestliže jsou silně závislé na počátečním stavu. (Příklad kuličky, která se vždy dokutálí do důlku, jestliže její počáteční stav je takový, že se "nedokutálí" jinam.) Tyto podivné atraktory se vyznačují chaotickou dynamikou a jsou fraktální. To znamená, že množina stavů atraktoru s okolím přitažlivosti zobrazená jako množina bodů vytváří zajímavé fraktální útvary. Nejednodušším a z hlediska fraktální geometrie zajímavým dynamickým systémem je nelineární systém s parabolickou funkcí

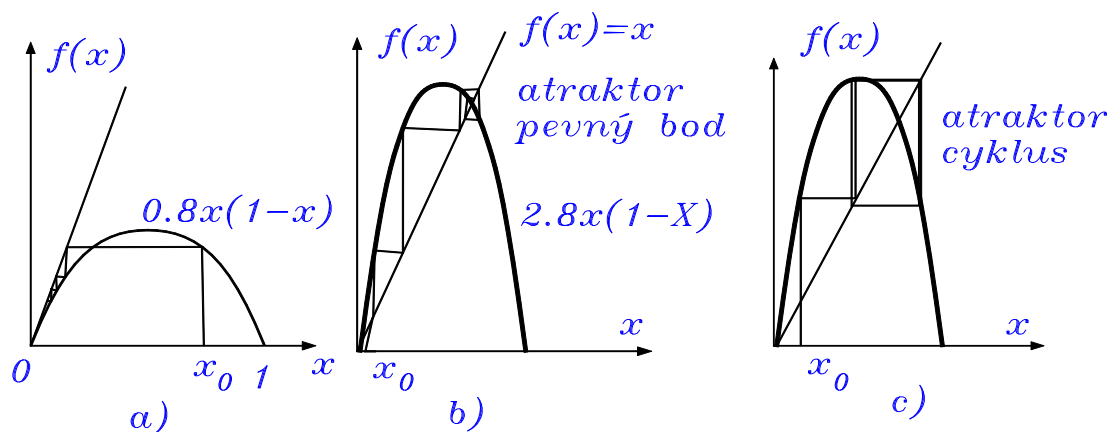
$$f(x) = \lambda \cdot x \cdot (1 - x), \text{ kde } 0 \leq x \leq 1.$$

Tímto dynamickým systémem můžeme na příklad modelovat růst populace určitého živočišného druhu. Chování systému pro $x \in (0, 1)$ závisí na hodnotě λ takto:

1. $\lambda \leq 1$ - dráhy všech bodů konvergují k nule. Obrázek 10.9a.
2. $1 < \lambda \leq 3$ - dráhy všech bodů konvergují k pevnému bodu zobrazení, pro který platí $f_\lambda(x) = x$. Obrázek 10.9b.
3. $\lambda > 3$ - dráha se ustálí podle velikosti λ v cyklu délky 2, 4, 16 ... Obrázek 10.9c.

Dráha bodu dynamického systému se zobrazením

$$f_{\lambda}(x) = \lambda \cdot x(1-x) \quad \text{pro } \lambda = 0.8, 2.8 \text{ a } 3.2.$$



Obr. 10.9

Označíme-li λ_n hodnotu, pro kterou bude mít cyklus délku 2^n . Lze dokázat, že posloupnost λ_n je konvergentní.

Označme $\lambda_{\infty} = \lim \lambda_n$. Je to hodnota parametru λ , při kterém v dráze bodu vzniká atraktor nekonečný cyklus. Až do této hodnoty se choval dynamický proces deterministicky vzhledem k počáteční hodnotě x_0 . To znamená, že bylo možné z přibližné znalosti počátečního stavu s jistotou předvídat její dlouhodobý časový vývoj. Teprve pro hodnoty $\lambda > \lambda_{\infty}$ se v systému objeví podivný atraktor. Jsou zde oblasti chaosu, kde je stav závislý na počáteční hodnotě x_0 a tyto oblasti se střídají s pravidelnými cykly.

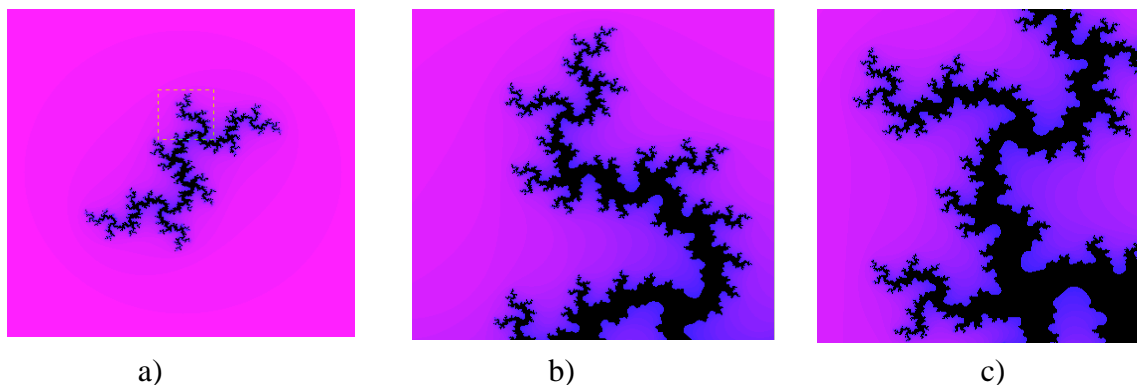
10.6. Juliova množina

Zvláště zajímavé fraktální množiny lze vygenerovat, jestliže přejdeme do oboru komplexních hodnot a zvolíme zobrazení

$$f_k(z) = z^2 + k, \quad z, k \in \mathbb{C},$$

kde k je komplexní konstanta a \mathbb{C} je Riemannova rovina komplexních čísel je atraktorem bod ∞ . Oblast přitažlivosti tohoto atraktoru má vždy hranici (existují body, jejichž dráhy nekonvergují k ∞). Hranice přitažlivosti bodu ∞ se nazývá **Juliova množina**.

Na obrázku 10.8 tvoří Juliovy množiny rozhraní mezi bílou (fialovou) a černou plochou. Bílá (fialová) plocha je oblast přitažlivosti bodu ∞ . Pouze pro $C = 0$ a $C = -2$ jsou Juliovy množiny jednoduché. Pro všechny ostatní hodnoty C jsou Juliovy množiny krásné fraktální množiny.



Obr. 10.8

Gaston Julia a Pierre Fatau publikovali v roce 1918 algoritmus pro konkrétní hodnoty k a různé počáteční podmínky (z). Pro interval $\langle -2 - 2i \rangle * \langle 2 + 2i \rangle$ se pro každý bod zkoumá chování interaktivního dynamického procesu.

System se prohlásí

- za stabilní, .. z_i se nedostane ze vzdálenosti kruhu o poloměru 2 pro pevný počet iterací;
- za nestabilní, pokud je po určitém počtu kroků $|z_i| \geq 2$.

V prvním případě se takovému bodu přiřadí barva. Ve druhém se takovému bodu přiřadí taková barva, za jak dlouho opustí kruh o poloměru 2. To je podle toho, jak rychle se vzdálí od počátečních podmínek.

program JULIA;

var z, k : Komplex;

Krok : **real;** {počet iterací cyklu v intervalu }

Dalsi : **real;** {krok k dalšímu bodu na obrazovce }

begin

$k . Re :=$ Pocatecni_stav . Re; {inicializace počátečních }

$k . In :=$ Pocatecni_stav . Im; {podmínek }

$z . Re := -2$; {malujeme v intervalu $\langle -2 - 2i \rangle * \langle 2 + 2i \rangle$ }

while $z . Re <> 2$ do

begin

$z . Im := -2$;

while $z . Im <> 2$ do

begin

Krok := 0; {testuj počet kroků }

```

    repeat      z := iterace (z,k);
    until (Krok >=Max) or (modul (z) >= 2 )
    namalu_bod (z.Re, z.Im, Krok div Max_barva);
    z.Im := z.Im + Dalsi
end;           { while pro z.Im }
z.Re := z.Re + Dalsi
end           { while pro z.Re }
end;         {JULIA }

```

10.7. Mandelbrotova množina

Mandelbrotovu množinu lze dostat jednoduchou modifikací předcházejícího algoritmu Julia.

V roce 1980 B. Mandelbrot hledal takový parametr k , aby $z = 0$.

program Mandelbrot;

var z, k : Komplex;

Krok : **real**; {počet iterací cyklu v intervalu }

Dalsi : **real**; {krok k dalšímu bodu na obrazovce }

begin

k . Re := -2; {malujeme v intervalu $\langle -2 - 2i \rangle * \langle 2 + 2i \rangle$ }

while k . Re \langle 2 do

begin

k . Im := -2;

while k.Im \langle 2 do

begin

z . RE := 0;

z . Im := 0;

Krok := 0;

repeat

z := iterace (z,k); {z := z * z+ k }

until (Krok >=Max) or (modul (z) >= 2)

namaluj_bod (z.Re, z.Im, Krok div Max_barva);

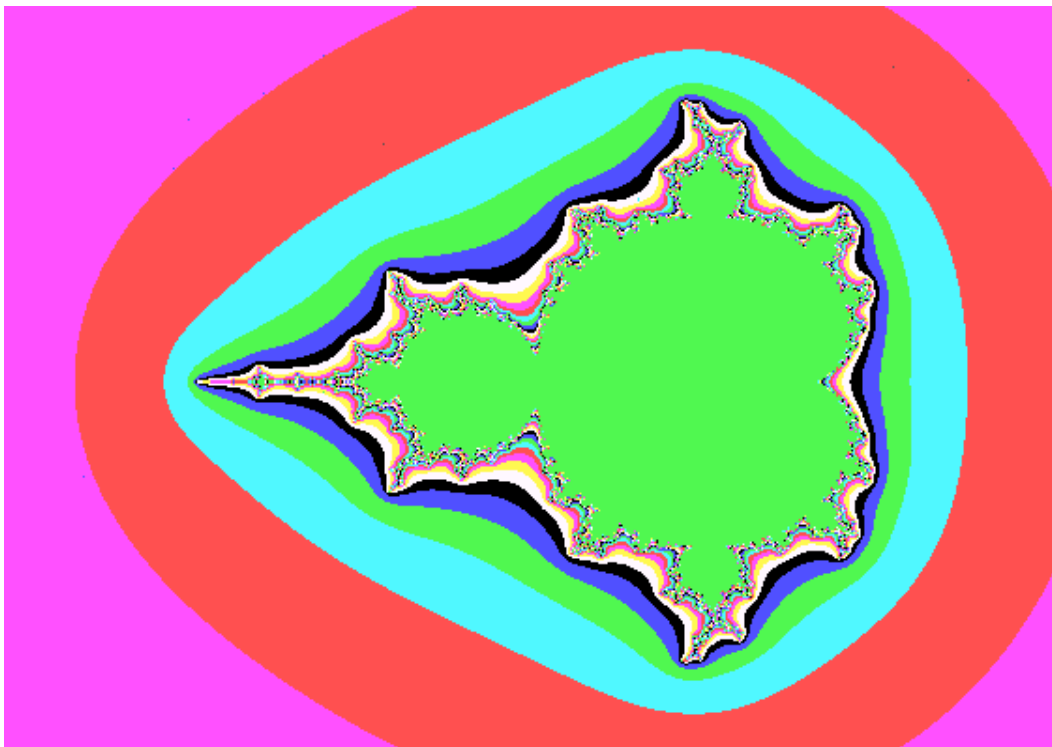
k.Im := k.Im + Dalsi

end; { while pro k.Im }

```

k.Re := k.Re + Dalsi
end                                     {while pro k.Re }
end.                                   {Mandelbrot }

```



Obr. 10.9

Na obrázku Obr. 10.9 je zobrazeno řešení pro interval: $\langle -2 - 2i \rangle * \langle 2 + 2i \rangle$.



Kontrolní otázky 10.

1. Vysvětlete pojmem fraktální geometrie?
2. Čím se liší fraktální geometrie při generování pravidelných útvarů od generování fraktálních útvarů, které se přibližují přírodě?
3. Napište - vysvětlete proceduru - program - pro kresbu pravidelného fraktálu.
Na př. sněhové vločky.
4. Napište - vysvětlete proceduru - program - pro kresbu nepravidelného fraktálu.
Na př. Brownův pohyb.