



Iterační princip - základ numerických výpočtů

Josef Daněk

Katedra matematiky
Fakulta aplikovaných věd
Západočeská univerzita

- motivace
- metody řešení nelineárních rovnic
 - metody: půlení intervalu, prosté iterace, Newtonova
 - zastavovací podmínky, konvergence, odhad chyby
- úloha na vlastní čísla matice
 - mocninná metoda
- Juliovy množiny
- Mandelbrotova množina
- deterministický chaos
- měření délky pobřeží
 - Kochova křivka, fraktální dimenze křivky

Motivace

Na **iteračním principu** je založeno velké množství metod ať z oblasti numerické matematiky, tak z celé řady dalších oblastí matematiky.

Vezměme si například polynom n -tého stupně:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0, \quad a_k \in \mathbb{C}, \quad k = 0, 1, \dots, n, \quad a_n \neq 0.$$

Polynom má právě n kořenů v $\mathbb{C}$ počítáme-li je i s jejich násobností.

Důkaz podal Jean-Robert Argand (1806).

Uvažujme nejjednodušší případy:

• lineární rovnice $ax + b = 0$ má řešení $x_1 = -\frac{b}{a}$ ✓

• kvadratická rovnice $ax^2 + bx + c = 0$

má řešení $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ (Babyloňané 2000 let př.n.l.) ✓

Motivace

- kubická rovnice $ax^3 + bx^2 + cx + d = 0$ má řešení, kde

$$x_1 = \sqrt[3]{\left(\frac{bc}{6a^2} - \frac{b^3}{27a^3} - \frac{d}{2a}\right) + \sqrt{\left(\frac{d}{2a} + \frac{b^3}{27a^3} - \frac{bc}{6a^2}\right)^2 + \left(\frac{c}{3a} - \frac{b^2}{9a^2}\right)^3}} - \dots$$

$x_{2,3} = \dots$ Niccolo Tartaglia, Scipione del Ferro (okolo 1500) ✓

- kvartická rovnice $ax^4 + bx^3 + cx^2 + dx + e = 0$ má řešení

$x_{1,2,3,4} = \dots$ Gerolamo Cardano, Lodovico Ferrari (1540) ✓

- pro polynom stupně $n \geq 5$ $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$

bylo dokázáno, že obecně nelze řešení spočítat pomocí konečného počtu kroků pomocí operací $+$, $-$, $\cdot$, $/$ a libovolné odmocniny $\sqrt[r]{}$, $r \in \mathbb{N}$

Paolo Ruffini (nekompletní 1813), Niels Henrik Abel (1824) ✗

Příklad 1

Hledání kořenů polynomu $P(x) = (x - 1)^n$.

Implementace v Matlabu (například pro $n = 4$)

```
%-----  
polynom=poly([ 1 1 1 1 ])  
koreny=roots(polynom)  
%-----  
  
polynom =    1    -4     6    -4     1  
  
koreny =  
  
1.000223371630863e+00  
9.999999706778071e-01 + 2.233423015147889e-04i  
9.999999706778071e-01 - 2.233423015147889e-04i  
9.997766870135252e-01
```

Metody řešení nelineárních rovnic

Příklad 2

Najděte kladné řešení rovnice $x^2 + x - 2 = 0$.

- 1. způsob:

úlohu lze řešit použitím vzorce pro řešení kvadratické rovnice

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

v našem případě vyjde:

$$x_{1,2} = \frac{-1 \pm \sqrt{1 + 4 \cdot 2}}{2 \cdot 1} = \frac{-1 \pm 3}{2} = \begin{cases} 1 \\ -2 \end{cases}$$

Odpověď: Kladné řešení rovnice je $\alpha = 1$.

Tento výpočet patří mezi **přímé metody**, tj. teoreticky po konečném počtu operací nalezneme přesné řešení. Je třeba si uvědomit, že jen pro některé problémy máme k dispozici vzorec a lze je řešit přímo.

Metody řešení nelineárních rovnic

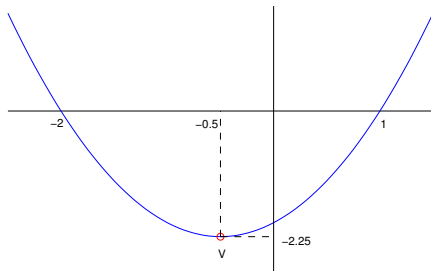
• 2. způsob:

pro řešení můžeme použít nějakou **iterační metodu**
jak vypadá graf funkce $y = x^2 + x - 2$?

$$y = \underbrace{\left(x + \frac{1}{2}\right)^2}_{x^2 + x + \frac{1}{4}} - \frac{1}{4} - 2$$

vrchol odpovídá bodu, ve kterém
je funkční hodnota minimální, tj.

pro $x = -\frac{1}{2}$ je $f\left(-\frac{1}{2}\right) = -2,25$



Princip iteračních metod spočívá v konstrukci posloupnosti
přibližných řešení tak, aby přesné řešení bylo limitním případem.
Musíme zadat **počáteční aproximaci** a iterační postup nějakým
způsobem **ukončit** (po konečném počtu kroků).

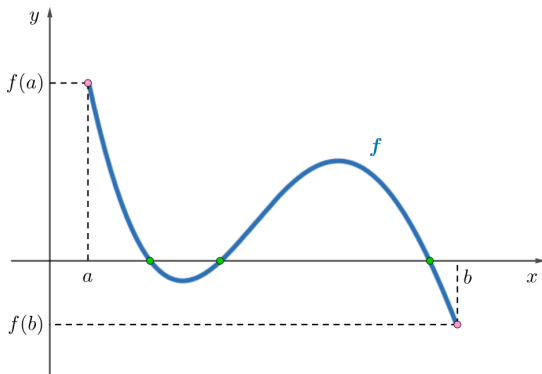
Metody řešení nelineárních rovnic

Bolzanova věta:

Pro reálnou funkci f předpokládejme:

- f je spojitá na uzavřeném intervalu $\langle a; b \rangle$,
- $f(a) \cdot f(b) < 0$.

Potom na $\langle a; b \rangle$ existuje aspoň jedno řešení rovnice $f(x) = 0$.



Metody řešení nelineárních rovnic

Metoda půlení intervalu (bisekce)

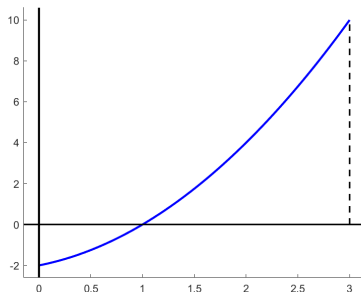
Algoritmus:

- 1 Zadáme $\varepsilon > 0$, a , b
- 2 $s = (a + b)/2$
- 3 Je-li $f(s) = 0$, pak $x = s$, KONEC
- 4 Je-li $f(s) \neq 0$, pak
 je-li $f(a) \cdot f(s) < 0$, pak $b = s$
 jinak $a = s$
- 5 Je-li $b - a \geq \varepsilon$, pak jdi na 2)
 jinak $x = (a + b)/2$, KONEC

Metody řešení nelineárních rovnic

Metoda pulení intervalu pro řešení nelineární rovnice $f(x)=0$ na intervalu $(0,3)$ a zastavovací podmínku $b-a < 0.1$

$$f(x) = x^2 + x - 2$$

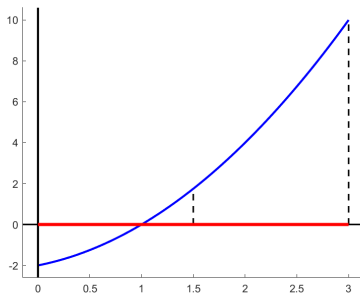


krok	a	b	s	f(a)	f(b)	f(s)

Metody řešení nelineárních rovnic

Metoda pulení intervalu pro řešení nelineární rovnice $f(x)=0$ na intervalu $\langle 0,3 \rangle$ a zastavovací podmínku $b-a < 0.1$

$$f(x) = x^2 + x - 2$$

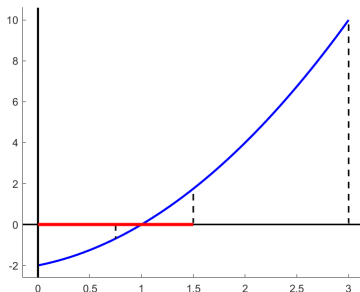


krok	a	b	s	f(a)	f(b)	f(s)
0	0.0000	3.0000	1.5000	-2.0000	10.0000	1.7500

Metody řešení nelineárních rovnic

Metoda pulení intervalu pro řešení nelineární rovnice $f(x)=0$ na intervalu $\langle 0,3 \rangle$ a zastavovací podmínku $b-a < 0.1$

$$f(x) = x^2 + x - 2$$

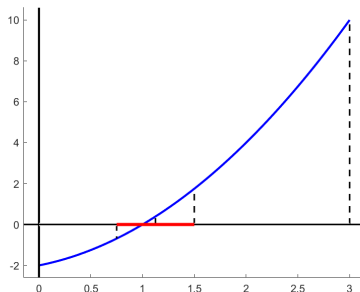


krok	a	b	s	f(a)	f(b)	f(s)
0	0.0000	3.0000	1.5000	-2.0000	10.0000	1.7500
1	0.0000	1.5000	0.7500	-2.0000	1.7500	-0.6875

Metody řešení nelineárních rovnic

Metoda pulení intervalu pro řešení nelineární rovnice $f(x)=0$ na intervalu $(0,3)$ a zastavovací podmínku $b-a < 0.1$

$$f(x) = x^2 + x - 2$$

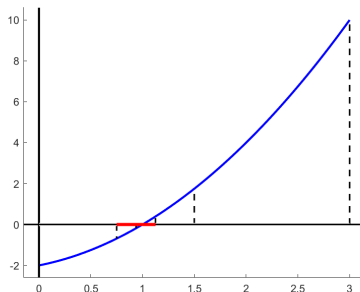


krok	a	b	s	f(a)	f(b)	f(s)
0	0.0000	3.0000	1.5000	-2.0000	10.0000	1.7500
1	0.0000	1.5000	0.7500	-2.0000	1.7500	-0.6875
2	0.7500	1.5000	1.1250	-0.6875	1.7500	0.3906

Metody řešení nelineárních rovnic

Metoda pulení intervalu pro řešení nelineární rovnice $f(x)=0$ na intervalu $\langle 0,3 \rangle$ a zastavovací podmínku $b-a < 0.1$

$$f(x) = x^2 + x - 2$$

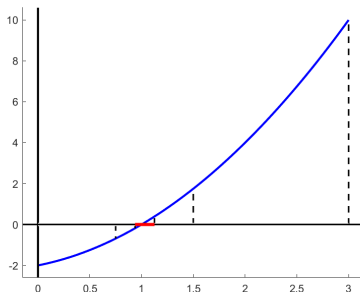


krok	a	b	s	f(a)	f(b)	f(s)
0	0.0000	3.0000	1.5000	-2.0000	10.0000	1.7500
1	0.0000	1.5000	0.7500	-2.0000	1.7500	-0.6875
2	0.7500	1.5000	1.1250	-0.6875	1.7500	0.3906
3	0.7500	1.1250	0.9375	-0.6875	0.3906	-0.1836

Metody řešení nelineárních rovnic

Metoda pulení intervalu pro řešení nelineární rovnice $f(x)=0$ na intervalu $\langle 0,3 \rangle$ a zastavovací podmínku $b-a < 0.1$

$$f(x) = x^2 + x - 2$$

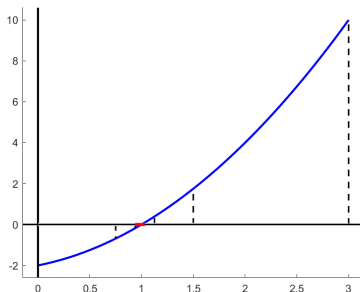


krok	a	b	s	f(a)	f(b)	f(s)
0	0.0000	3.0000	1.5000	-2.0000	10.0000	1.7500
1	0.0000	1.5000	0.7500	-2.0000	1.7500	-0.6875
2	0.7500	1.5000	1.1250	-0.6875	1.7500	0.3906
3	0.7500	1.1250	0.9375	-0.6875	0.3906	-0.1836
4	0.9375	1.1250	1.0313	-0.1836	0.3906	0.0947

Metody řešení nelineárních rovnic

Metoda pulení intervalu pro řešení nelineární rovnice $f(x)=0$ na intervalu $(0,3)$ a zastavovací podmínku $b-a < 0.1$

$$f(x) = x^2 + x - 2$$



krok	a	b	s	f(a)	f(b)	f(s)
0	0.0000	3.0000	1.5000	-2.0000	10.0000	1.7500
1	0.0000	1.5000	0.7500	-2.0000	1.7500	-0.6875
2	0.7500	1.5000	1.1250	-0.6875	1.7500	0.3906
3	0.7500	1.1250	0.9375	-0.6875	0.3906	-0.1836
4	0.9375	1.1250	1.0313	-0.1836	0.3906	0.0947
5	0.9375	1.0313	0.9844	-0.1836	0.0947	-0.0466

Metody řešení nelineárních rovnic

Metoda prosté iterace

Algoritmus:

- 1 Zadáme $x_0 \in \langle a; b \rangle$, $\varepsilon > 0$
- 2 $x_{k+1} = \varphi(x_k)$
- 3 Je-li $|x_{k+1} - x_k| < \varepsilon$,
pak $x = x_{k+1}$, KONEC
jinak jdi na 2)

Metody řešení nelineárních rovnic

Metoda prosté iterace pro řešení nelineární rovnice $x = \phi(x)$
pro počáteční aproximaci $x_0 = 1.5$ a zast. podmínku $|x(k) - x(k-1)| < 0.1$

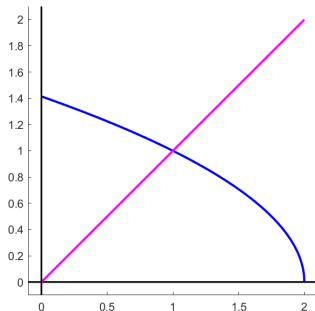
$$x^2 + x - 2 = 0$$

$$x^2 = 2 - x$$

$$x = \sqrt{2 - x}$$

tj. $\phi(x) = \sqrt{2 - x}$

	krok		$x(k)$		$x(k) - x(k-1)$	
--	------	--	--------	--	-----------------	--



Metody řešení nelineárních rovnic

Metoda prosté iterace pro řešení nelineární rovnice $x = \phi(x)$
pro počáteční aproximaci $x_0 = 1.5$ a zast. podmínku $|x(k) - x(k-1)| < 0.1$

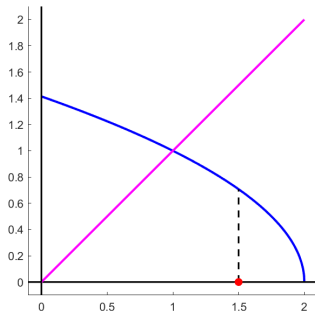
$$x^2 + x - 2 = 0$$

$$x^2 = 2 - x$$

$$x = \sqrt{2 - x}$$

tj. $\phi(x) = \sqrt{2 - x}$

krok	$x(k)$	$ x(k) - x(k-1) $
0	1.500000	



Metody řešení nelineárních rovnic

Metoda prosté iterace pro řešení nelineární rovnice $x = \phi(x)$
pro počáteční aproximaci $x_0 = 1.5$ a zast. podmínku $|x(k) - x(k-1)| < 0.1$

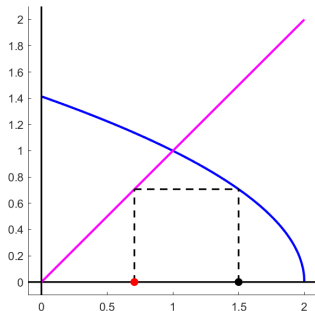
$$x^2 + x - 2 = 0$$

$$x^2 = 2 - x$$

$$x = \sqrt{2 - x}$$

tj. $\phi(x) = \sqrt{2 - x}$

krok	$x(k)$	$ x(k) - x(k-1) $
0	1.500000	
1	0.707107	0.792893



Metody řešení nelineárních rovnic

Metoda prosté iterace pro řešení nelineární rovnice $x = \phi(x)$
pro počáteční aproximaci $x_0 = 1.5$ a zast. podmínku $|x(k) - x(k-1)| < 0.1$

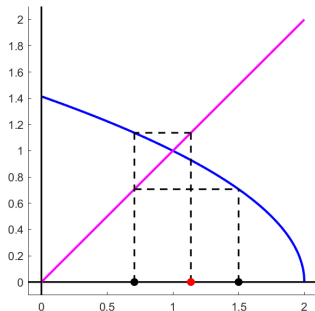
$$x^2 + x - 2 = 0$$

$$x^2 = 2 - x$$

$$x = \sqrt{2 - x}$$

tj. $\phi(x) = \sqrt{2 - x}$

krok	$x(k)$	$ x(k) - x(k-1) $
0	1.500000	
1	0.707107	0.792893
2	1.137055	0.429948



Metody řešení nelineárních rovnic

Metoda prosté iterace pro řešení nelineární rovnice $x = \phi(x)$
pro počáteční aproximaci $x_0 = 1.5$ a zast. podmínku $|x(k) - x(k-1)| < 0.1$

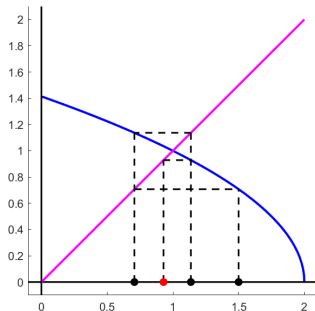
$$x^2 + x - 2 = 0$$

$$x^2 = 2 - x$$

$$x = \sqrt{2 - x}$$

tj. $\phi(x) = \sqrt{2 - x}$

krok	$x(k)$	$ x(k) - x(k-1) $
0	1.500000	
1	0.707107	0.792893
2	1.137055	0.429948
3	0.928949	0.208106



Metody řešení nelineárních rovnic

Metoda prosté iterace pro řešení nelineární rovnice $x = \phi(x)$
pro počáteční aproximaci $x_0 = 1.5$ a zast. podmínku $|x(k) - x(k-1)| < 0.1$

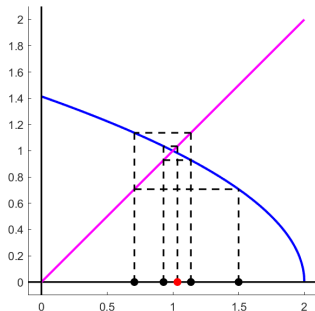
$$x^2 + x - 2 = 0$$

$$x^2 = 2 - x$$

$$x = \sqrt{2 - x}$$

tj. $\phi(x) = \sqrt{2 - x}$

krok	$x(k)$	$ x(k) - x(k-1) $
0	1.500000	
1	0.707107	0.792893
2	1.137055	0.429948
3	0.928949	0.208106
4	1.034916	0.105968



Metody řešení nelineárních rovnic

Metoda prosté iterace pro řešení nelineární rovnice $x = \phi(x)$
pro počáteční aproximaci $x_0 = 1.5$ a zast. podmínku $|x(k) - x(k-1)| < 0.1$

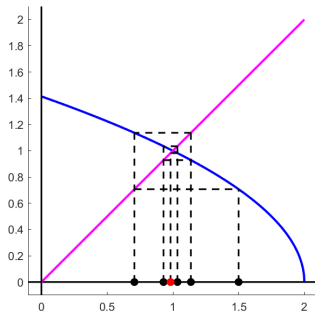
$$x^2 + x - 2 = 0$$

$$x^2 = 2 - x$$

$$x = \sqrt{2 - x}$$

tj. $\phi(x) = \sqrt{2 - x}$

krok	$x(k)$	$ x(k) - x(k-1) $
0	1.500000	
1	0.707107	0.792893
2	1.137055	0.429948
3	0.928949	0.208106
4	1.034916	0.105968
5	0.982387	0.052529



Metody řešení nelineárních rovnic

Newtonova metoda (metoda tečen)

Algoritmus:

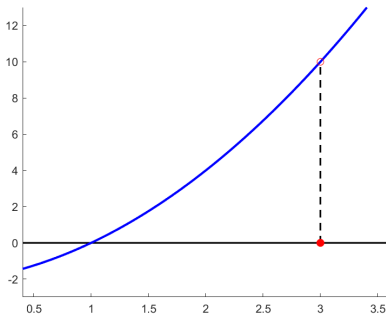
1. Zadáme $x_0 \in \langle a; b \rangle$, $\varepsilon > 0$
2.
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$
3. Je-li $|x_{k+1} - x_k| < \varepsilon$,
pak $x = x_{k+1}$, KONEC
jinak jdi na 2)

Metody řešení nelineárních rovnic

Newtonova metoda pro řešení nelineární rovnice $f(x)=0$
pro počáteční aproximaci $x_0=3$ a zastavovací podmínku $|x(k)-x(k-1)| < 0.1$

$$f(x) = x^2 + x - 2$$

$$f'(x) = 2x + 1$$



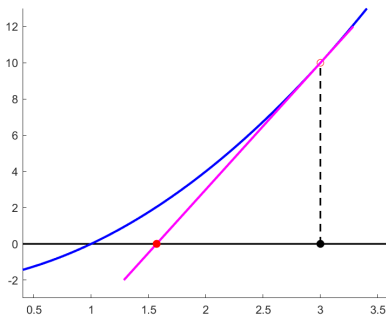
krok	$x(k)$	$ x(k) - x(k-1) $	$f(x(k))$
0	3.00000000		

Metody řešení nelineárních rovnic

Newtonova metoda pro řešení nelineární rovnice $f(x)=0$
pro počáteční aproximaci $x_0=3$ a zastavovací podmínku $|x(k)-x(k-1)| < 0.1$

$$f(x) = x^2 + x - 2$$

$$f'(x) = 2x + 1$$



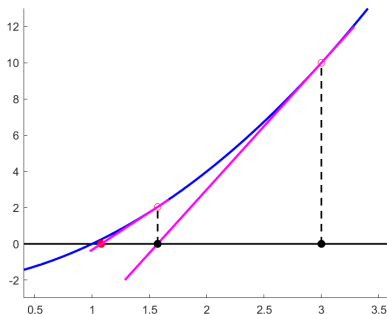
krok	$x(k)$	$ x(k) - x(k-1) $	$f(x(k))$
0	3.00000000		
1	1.57142857	1.42857143	2.04081633

Metody řešení nelineárních rovnic

Newtonova metoda pro řešení nelineární rovnice $f(x)=0$
pro počáteční aproximaci $x_0=3$ a zastavovací podmínku $|x(k)-x(k-1)| < 0.1$

$$f(x) = x^2 + x - 2$$

$$f'(x) = 2x + 1$$



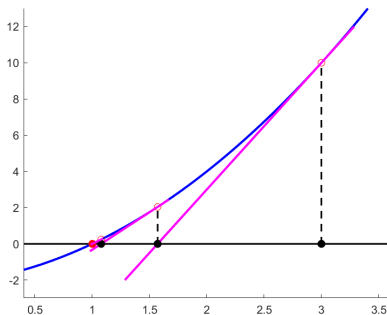
krok	$x(k)$	$ x(k) - x(k-1) $	$f(x(k))$
0	3.00000000		
1	1.57142857	1.42857143	2.04081633
2	1.07881773	0.49261084	0.24266544

Metody řešení nelineárních rovnic

Newtonova metoda pro řešení nelineární rovnice $f(x)=0$
pro počáteční aproximaci $x_0=3$ a zastavovací podmínku $|x(k)-x(k-1)| < 0.1$

$$f(x) = x^2 + x - 2$$

$$f'(x) = 2x + 1$$



krok	$x(k)$	$x(k) - x(k-1)$	$f(x(k))$
0	3.00000000		
1	1.57142857	1.42857143	2.04081633
2	1.07881773	0.49261084	0.24266544
3	1.00196737	0.07685036	0.00590598

Zastavovací podmínky

Iterační proces musíme nějakým způsobem ukončit.

Můžeme použít některou z následujících podmínek:

- počet kroků metody N
- délka intervalu $\langle a; b \rangle$ v daném kroku $b - a < \varepsilon$
- absolutní hodnota funkční hodnoty v bodě x_k $|f(x_k)| < \varepsilon$
- vzdálenost dvou posledních iterací x_k a x_{k-1} $|x_k - x_{k-1}| < \varepsilon$
- ...

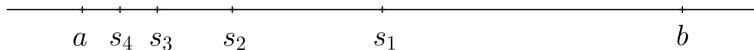
Jakou použít podmínku pro danou metodu?

Kdy zafunguje daná podmínka?

Jaká je souvislost zastavovací podmínky a přesného řešení?

Konvergence

- $b - a < \varepsilon$ (Bisekce)



$$(b - a) \cdot \left(\frac{1}{2}\right)^k < \varepsilon$$

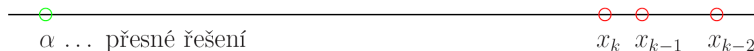
$$\frac{b - a}{\varepsilon} < 2^k$$

$$k = \left\lceil \log_2 \frac{b - a}{\varepsilon} \right\rceil$$

př.: Kolik je třeba iterací na zmenšení intervalu na $\frac{1}{100}$ původní délky?

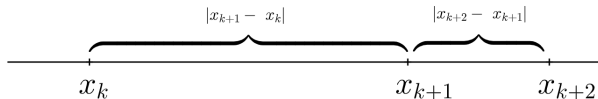
$$k = \lceil \log_2 100 \rceil = \left\lceil \frac{\log 100}{\log 2} \right\rceil = 7$$

- $|x_k - x_{k-1}| < \varepsilon$ (Metoda prosté iterace, Newtonova metoda, ...)



Konvergence

Posloupnost iterací bude konvergovat, pokud se iterace **zahušťují**.



Pro každé $k (> k_0)$ musí platit:

$$|x_{k+2} - x_{k+1}| < |x_{k+1} - x_k|.$$

V metodě prosté iterace používáme iterační formuli $x_{k+1} = \varphi(x_k)$, tj.

$$|\varphi(x_{k+1}) - \varphi(x_k)| < |x_{k+1} - x_k|.$$

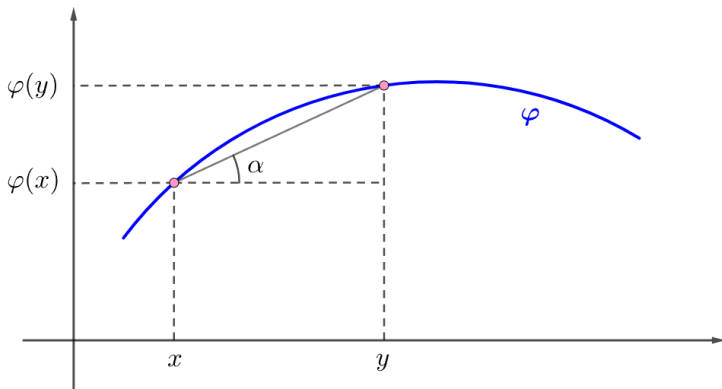
Po vydělení $|x_{k+1} - x_k| \neq 0$ dostaneme

$$\frac{|\varphi(x_{k+1}) - \varphi(x_k)|}{|x_{k+1} - x_k|} < 1 \quad (\star)$$

Konvergence

(★) je splněno, pokud je $\varphi = \varphi(x)$ **kontrakce** (má omezený růst), tj.

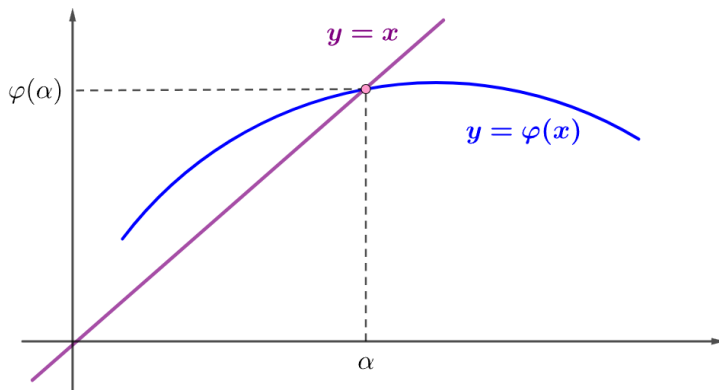
$$\frac{|\varphi(x) - \varphi(y)|}{|x - y|} \leq q < 1 \quad \forall x, y \in I$$



Konvergence

V metodě prosté iterace hledáme **pevný bod**.

Přesné řešení α musí splňovat rovnost $\alpha = \varphi(\alpha)$.

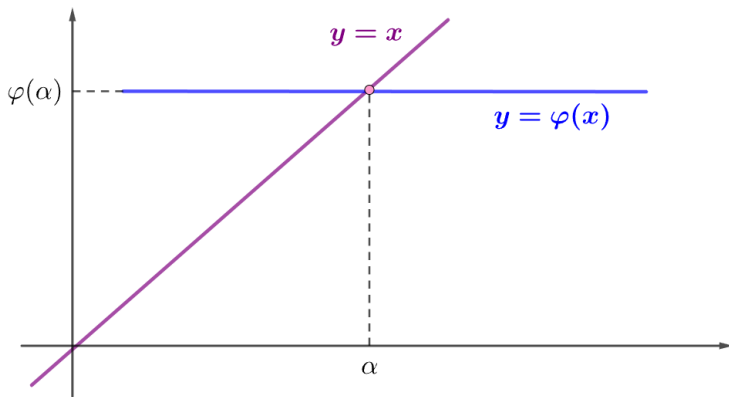


Konvergence

Pokud by byla funkce $\varphi = \varphi(x)$ konstantní $\Rightarrow$

$\Rightarrow$ metoda prosté iterace najde přesné řešení v jedné iteraci:

$$x_1 = \varphi(x_0) = \text{konst.} = \varphi(\alpha) = \alpha$$



Odhad chyby

- Pro **metodu půlení intervalu** ... zřejmé.

- Pro **metodu prosté iterace**:

- Máme konvergentní proces $x_k = \varphi(x_{k-1}), \quad k = 1, 2, \dots$

- Přesné řešení α splňuje vztah $\alpha = \varphi(\alpha), \quad \lim_{k \rightarrow \infty} x_k = \alpha$

- Po odečtení dostaneme $x_k - \alpha = \varphi(x_{k-1}) - \varphi(\alpha)$

$$\text{tj. } |x_k - \alpha| = |\varphi(x_{k-1}) - \varphi(\alpha)| \quad \leftarrow$$

- φ je kontrakce $|\varphi(x_{k-1}) - \varphi(\alpha)| \leq q \cdot |x_{k-1} - \alpha| \quad \leftarrow$

- Dále použijeme $\triangle$ nerovnost:

$$|x_{k-1} - \alpha| = |x_{k-1} - x_k + x_k - \alpha| \leq |x_{k-1} - x_k| + |x_k - \alpha| \quad \leftarrow$$

- Dostaneme: $|x_k - \alpha| \leq q \cdot |x_{k-1} - x_k| + q \cdot |x_k - \alpha|$

$$(1 - q) \cdot |x_k - \alpha| \leq q \cdot |x_{k-1} - x_k| \quad / \cdot \frac{1}{1-q} > 0$$

$$|x_k - \alpha| \leq \frac{q}{1-q} \cdot |x_{k-1} - x_k|$$

Odhad chyby

Příklad 3 Uvažujme iterační formuli $x_{k+1} = 0,99 x_k + 1, \quad x_0 = 0$.

růst funkce vyjadřuje koeficient v lineárním členu $\Rightarrow q = 0,99$

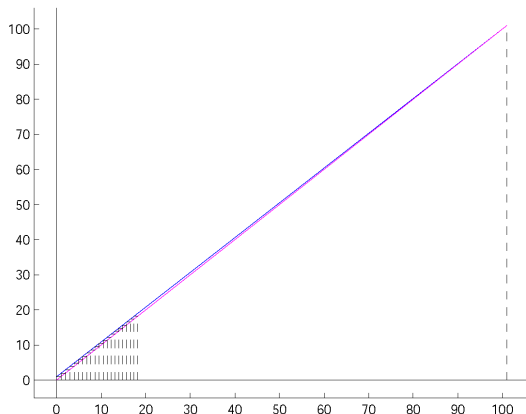
odhad chyby: $|x_k - \alpha| \leq \frac{0,99}{1-0,99} |x_{k-1} - x_k| = 99 |x_{k-1} - x_k|$

k	x (k)	dx (k) = x (k) - x (k-1)	dx (k) / dx (k-1)
0	0.000000		
1	1.000000	1.000000	
2	1.990000	0.990000	0.990000
3	2.970100	0.980100	0.990000
4	3.940399	0.970299	0.990000
5	4.900995	0.960596	0.990000
6	5.851985	0.950990	0.990000
7	6.793465	0.941480	0.990000
8	7.725531	0.932065	0.990000
18	16.548624	0.842943	0.990000
19	17.383138	0.834514	0.990000
20	18.209306	0.826169	0.990000

Odhad chyby

Jaký je odhad chyby x_{20} ?

$$|x_{20} - \alpha| \leq \frac{0,99}{1 - 0,99} |x_{20} - x_{19}| = 99 |x_{20} - x_{19}| = 99 \cdot 0,826169 \doteq 81,8$$



přesné řešení α :

$$\alpha = 0,99 \cdot \alpha + 1$$

$$0,01 \cdot \alpha = 1$$

$$\alpha = 100$$

Odhad chyby

Příklad 4 Uvažujme iterační formuli $x_{k+1} = 0,1 x_k + 1, \quad x_0 = 0$.

růst funkce vyjadřuje koeficient v lineárním členu $\Rightarrow q = 0,1$

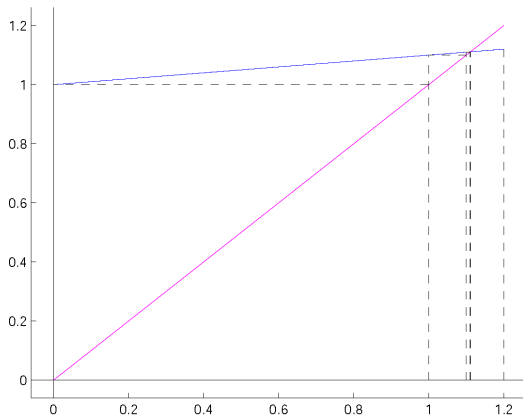
odhad chyby: $|x_k - \alpha| \leq \frac{0,1}{1-0,1} |x_{k-1} - x_k| = \frac{1}{9} |x_{k-1} - x_k|$

k	x (k)	dx (k) = x (k) - x (k-1)	dx (k) / dx (k-1)
0	0.000000		
1	1.000000	1.000000	
2	1.100000	0.100000	0.100000
3	1.110000	0.010000	0.100000
4	1.111000	0.001000	0.100000

Odhad chyby

Jaký je odhad chyby x_4 ?

$$|x_4 - \alpha| \leq \frac{0,1}{1 - 0,1} |x_4 - x_3| = \frac{1}{9} |x_4 - x_3| = \frac{1}{9} \cdot 0,001 = 1,1 \cdot 10^{-4} \approx 0,0001$$



přesné řešení α :

$$\alpha = 0,1 \cdot \alpha + 1$$

$$0,9 \cdot \alpha = 1$$

$$\alpha = \frac{10}{9} = 1,1$$

Newtonova metoda v komplexním oboru

Algoritmus Newtonovy metody můžeme použít pro řešení rovnice v $\mathbb{C}$.

Příklad 5

Newtonovou metodou řešte v komplexním oboru rovnici

$$z^3 - 1 = 0, \quad z = x + iy$$

Iterační formule:

$$z_{k+1} = z_k - \frac{z_k^3 - 1}{3z_k^2}$$

Úloha má 3 řešení: $1, \quad -\frac{1}{2} + \frac{\sqrt{3}}{2}i, \quad -\frac{1}{2} - \frac{\sqrt{3}}{2}i$

Volbou konkrétní počáteční aproximace ze čtverce $\langle -2; 2 \rangle \times \langle -2; 2 \rangle$ dostaneme jedno uvedených řešení.

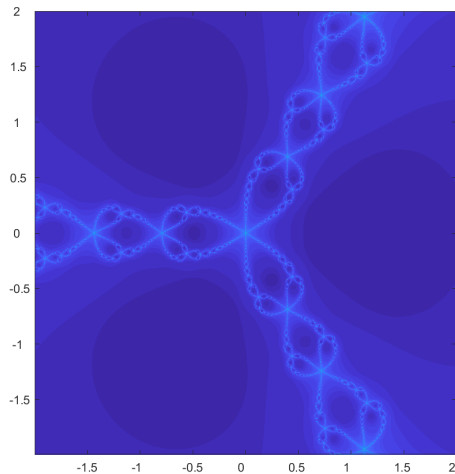
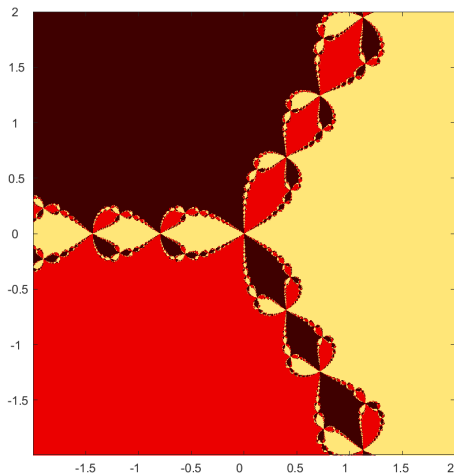
Počáteční aproximace obarvíme podle:

- toho, ke kterému řešení dospějeme
- podle počtu iterací potřebných k rozhodnutí

Získáme **fraktálovou strukturu**.

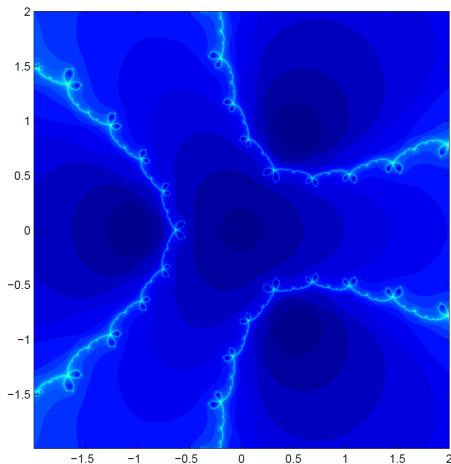
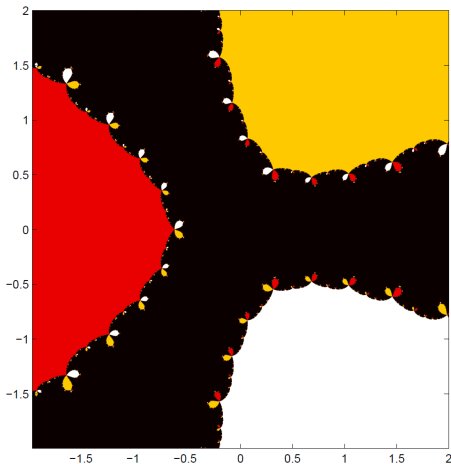
Newtonova metoda v komplexním oboru

$$z^3 - 1 = 0$$



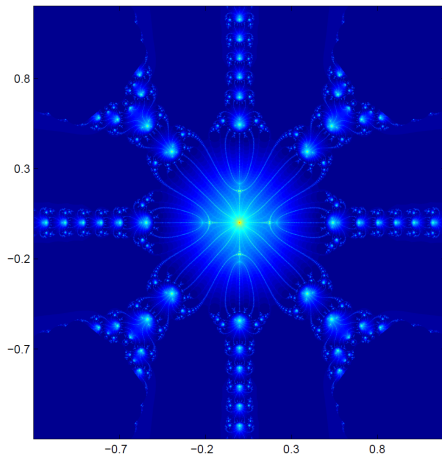
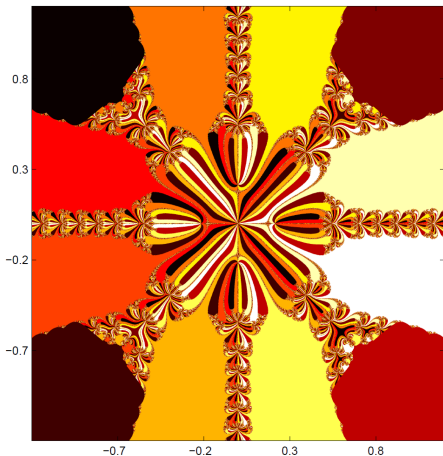
Newtonova metoda v komplexním oboru

$$z^4 + z = 0$$



Newtonova metoda v komplexním oboru

$$z^{12} + \frac{744}{611} z^8 - \frac{86}{16057} z^4 + \frac{25}{357} = 0$$



Úloha na vlastní čísla

S úlohou na vlastní čísla se setkáme v aplikacích při řešení celé řady technických a fyzikálních problémů. Vlastní čísla hrají důležitou roli například u iteračních metod pro řešení soustavy lineárních rovnic. Rozhodují o konvergenci příslušné iterační metody.

Definice: $\mathbf{A}$ je čtvercová matice řádu n . Číslo λ , pro které má soustava

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \quad \text{resp.} \quad (\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$$

nenulové řešení, se nazývá **vlastní číslo** matice $\mathbf{A}$,
jemu odpovídající nenulové řešení $\mathbf{v}$ **vlastní vektor** matice $\mathbf{A}$.

Homogenní soustava má nenulové řešení $\Leftrightarrow$ matice je singulární,
tj. její determinant je nulový.

Vlastní čísla $\lambda_1, \lambda_2, \dots, \lambda_n$ jsou kořeny **charakteristické rovnice**

$$p_{\mathbf{A}}(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = 0.$$

Úloha na vlastní čísla

Příklad Určete vlastní čísla a vlastní vektory matice $\mathbf{A} = \begin{bmatrix} -1 & -3 \\ 0 & 2 \end{bmatrix}$

- $\lambda_1 = -1, v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

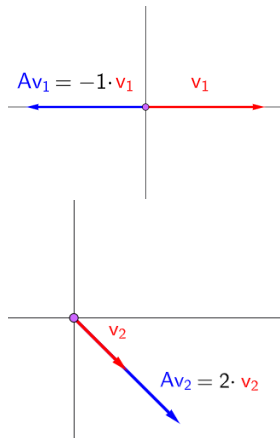
$$\mathbf{A}v_1 = \lambda_1 v_1$$

$$\begin{bmatrix} -1 & -3 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = -1 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- $\lambda_2 = 2, v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

$$\mathbf{A}v_2 = \lambda_2 v_2$$

$$\begin{bmatrix} -1 & -3 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 2 \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



Úloha na vlastní čísla

Charakteristický polynom je stupně $n \Rightarrow \exists n$ vlastních čísel.

Ke každému vlastnímu číslu λ_i existuje alespoň jeden vlastní vektor $\mathbf{v}_i$.

Jak bylo řečeno, v řadě problémů je důležité určit tzv. **dominantní vlastní číslo**, tj. vlastní číslo, které je největší v absolutní hodnotě.

Úlohu určit dominantní vlastní číslo nazýváme **částečným problémem** a jednou z metod, která jej řeší je **mocninná metoda**.

Předpoklady pro **mocninnou metodu**:

- 1 **A** má n -lineárně nezávislých vlastních vektorů
- 2 existuje jediné dominantní vlastní číslo
- 3 vlastní čísla lze seřadit: $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$.

Mocnná metoda

Algoritmus:

- 1 Zvolíme nenulový vektor $\mathbf{y}^{(0)}$.
- 2 Sestrojíme posloupnost $\mathbf{y}^{(k+1)} = \mathbf{A}\mathbf{y}^{(k)}$, $k = 0, 1, \dots$
- 3 Vybereme v absolutní hodnotě největší složku vektoru $\mathbf{y}^{(k+1)}$
a vydělíme ji stejnou složkou vektoru předchozí iterace $\mathbf{y}^{(k)}$
a určíme iteraci $\lambda_1^{(k+1)} = \frac{y_j^{(k+1)}}{y_j^{(k)}}$.
- 4 Iterujeme dokud není splněna podmínka $|\lambda_1^{(k+1)} - \lambda_1^{(k)}| < \varepsilon$.

Mocninná metoda

Náznak odvození:

- ❶ Zvolíme $\mathbf{y}^{(0)}$ jako lineární kombinaci vlastních vektorů

$$\mathbf{y}^{(0)} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_n \mathbf{v}_n.$$

- ❷ Sestrojíme posloupnost

$$\mathbf{y}^{(k)} = \mathbf{A} \mathbf{y}^{(k-1)}, \quad \text{tj.} \quad \mathbf{y}^{(k)} = \mathbf{A}^k \mathbf{y}^{(0)}.$$

$$\mathbf{y}^{(k)} = \alpha_1 \mathbf{A}^k \mathbf{v}_1 + \alpha_2 \mathbf{A}^k \mathbf{v}_2 + \cdots + \alpha_n \mathbf{A}^k \mathbf{v}_n.$$

- ❸ Platí $\mathbf{A} \mathbf{v}_i = \lambda_i \mathbf{v}_i$, proto

$$\mathbf{y}^{(k)} = \alpha_1 \lambda_1^k \mathbf{v}_1 + \alpha_2 \lambda_2^k \mathbf{v}_2 + \cdots + \alpha_n \lambda_n^k \mathbf{v}_n.$$

- ❹ Vytkneme dominantní vlastní číslo v $\mathbf{y}^{(k)}$ a analogicky v $\mathbf{y}^{(k+1)}$

$$\mathbf{y}^{(k+1)} = \lambda_1^{k+1} \left[\alpha_1 \mathbf{v}_1 + \underbrace{\sum_{i=2}^n \alpha_i \overbrace{\left(\frac{\lambda_i}{\lambda_1} \right)^{k+1}}^{\rightarrow 0}}_{\varepsilon_{k+1} \rightarrow 0} \mathbf{v}_i \right].$$

Mocninná metoda

Příklad Určete dominantní vlastní číslo matice

$$\mathbf{A} = \begin{bmatrix} 900 & 20 & 1 \\ 20 & 500 & 30 \\ 1 & 30 & 100 \end{bmatrix}.$$

k	y(1)_k	y(2)_k	y(3)_k	lambda_k
0	1.000000e+000	1.000000e+000	1.000000e+000	
1	9.210000e+002	5.500000e+002	1.310000e+002	921.0000000
2	8.400310e+005	2.973500e+005	3.052100e+004	912.0857763
3	7.620054e+008	1.663913e+008	1.281263e+007	907.1158338
4	6.891455e+011	9.882011e+010	7.035006e+009	904.3840077
5	6.222144e+014	6.340402e+013	4.357249e+012	902.8781109
6	5.612654e+017	4.427701e+016	2.960060e+015	902.0450147
7	5.060274e+020	3.345262e+019	2.185582e+018	901.5830307
8	4.560959e+023	2.691242e+022	1.728164e+021	901.3264854
9	4.110263e+026	2.262997e+025	1.436285e+024	901.1839104
10	3.703777e+029	1.957860e+028	1.233554e+027	901.1046393

vlastní čísla z MATLABu: [97,76221 501,23249 901,00529]

Poznámka:

Abychom zamezili přetečení (podtečení) je vhodné vektor $\mathbf{y}^{(k)}$ normovat.

Mocninná metoda

Příklad Určete dominantní vlastní číslo matice

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

k	y(1)_k	y(2)_k	y(3)_k	lambda_k
0	1.000000e+000	1.000000e+000	1.000000e+000	
1	5.000000e+000	0.000000e+000	1.000000e+000	5.0000000
2	1.600000e+001	-5.000000e+000	1.000000e+000	3.2000000
3	4.400000e+001	-2.100000e+001	1.000000e+000	2.7500000
4	1.120000e+002	-6.500000e+001	1.000000e+000	2.5454545
5	2.720000e+002	-1.770000e+002	1.000000e+000	2.4285714
10	1.638400e+004	-1.331300e+004	1.000000e+000	2.2068966
50	8.556839e+016	-8.219069e+016	1.000000e+000	2.0402685
100	1.914152e+032	-1.876123e+032	1.000000e+000	2.0200669
200	4.836884e+062	-4.788675e+062	1.000000e+000	2.0100167
300	9.187032e+092	-9.125921e+092	1.000000e+000	2.0066741
1000	1.608334e+304	-1.605120e+304	1.000000e+000	2.0020007

vlastní čísla z MATLABu: [2 2 1] a pouze 2 vlastní vektory

Poznámka:

Při nesplněných předpokladech odvození může být konvergence pomalá.

Mocninná metoda

Příklad Určete dominantní vlastní číslo matice

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}.$$

k	y(1)_k	y(2)_k	y(3)_k	1s	2s	3s
0	2.000000e+00	1.000000e+00	1.000000e+00			
1	3.000000e+00	2.000000e+00	3.000000e+00	1.5000000	2	3
2	5.000000e+00	4.000000e+00	9.000000e+00	1.6666667	2	3
3	9.000000e+00	8.000000e+00	2.700000e+01	1.8000000	2	3
4	1.700000e+01	1.600000e+01	8.100000e+01	1.8888889	2	3
5	3.300000e+01	3.200000e+01	2.430000e+02	1.9411765	2	3
6	6.500000e+01	6.400000e+01	7.290000e+02	1.9696970	2	3
7	1.290000e+02	1.280000e+02	2.187000e+03	1.9846154	2	3
8	2.570000e+02	2.560000e+02	6.561000e+03	1.9922481	2	3
9	5.130000e+02	5.120000e+02	1.968300e+04	1.9961089	2	3
10	1.025000e+03	1.024000e+03	5.904900e+04	1.9980507	2	3

vlastní čísla z MATLABu: [3 2 1] $\mathbf{v}_1 = [0, 0, 1]^T$

Poznámka:

Pro dělení je nejlépe vybírat maximální složku vektoru v abs. hodnotě, zejména v situaci, když má vlastní vektor některé složky nulové.

Mocninná metoda

Příklad Určete dominantní vlastní číslo matice

$$\mathbf{A} = \begin{bmatrix} 4 & -3 & 2 \\ 0 & 2 & 0 \\ 0 & -4 & 6 \end{bmatrix}.$$

k	y(1)_k	y(2)_k	y(3)_k	lambda_k
0	1.000000e+00	1.000000e+00	1.000000e+00	
1	3.000000e+00	2.000000e+00	2.000000e+00	3.0000000
2	1.000000e+01	4.000000e+00	4.000000e+00	3.3333333
3	3.600000e+01	8.000000e+00	8.000000e+00	3.6000000
4	1.360000e+02	1.600000e+01	1.600000e+01	3.7777778
5	5.280000e+02	3.200000e+01	3.200000e+01	3.8823529
6	2.080000e+03	6.400000e+01	6.400000e+01	3.9393939
7	8.256000e+03	1.280000e+02	1.280000e+02	3.9692308
8	3.289600e+04	2.560000e+02	2.560000e+02	3.9844961
9	1.313280e+05	5.120000e+02	5.120000e+02	3.9922179

vlastní čísla: $[6 \ 4 \ 2]$ $v_1 = [1, 0, 1]^T$, $v_2 = [1, 0, 0]^T$, $v_3 = [1, 2, 2]^T$

$$y_0 = 0 \cdot v_1 + 0,5 \cdot v_2 + 0,5 \cdot v_3$$

Poznámka:

Pokud ve zvolené počáteční aproximaci y_0 **není** obsažen vlastní vektor v_1 , mocninná metoda nevypočte dominantní vlastní číslo λ_1 .

Mocninná metoda

Příklad Určete dominantní vlastní číslo matice

$$\mathbf{A} = \begin{bmatrix} 3 & 3 & 0 \\ 0 & 4 & 2 \\ 0 & 0 & 1 \end{bmatrix}.$$

k	y (1)_k		y (2)_k		y (3)_k		lambda_k	
0	1.000000e+000		1.000000e+000		1.000000e+000			
1	6.000000e+000		6.000000e+000		1.000000e+000		6.0000000	
2	3.600000e+001		2.600000e+001		1.000000e+000		6.0000000	

vlastní čísla: $[4 \ 3 \ 1]$ $v_1 = [3, 1, 0]^T$, $v_2 = [1, 0, 0]^T$, $v_3 = [3, -2, 3]^T$

Poznámka:

$$y_0 = 1, \overline{6} \cdot v_1 - 5 \cdot v_2 + 0, \overline{3} \cdot v_3$$

Předpoklady metody jsou splněny i ve zvoleném počátečním vektoru y_0
je obsažen vlastní vektor v_1 , přesto metoda skončí v nesprávné hodnotě.

Mocninná metoda

Příklad Určete dominantní vlastní číslo matice

$$\mathbf{A} = \begin{bmatrix} 3 & 3 & 0 \\ 0 & 4 & 2 \\ 0 & 0 & 1 \end{bmatrix}.$$

k	y (1)_k	y (2)_k	y (3)_k	lambda_k
0	1.000000e+000	1.000000e+000	1.000000e+000	
1	6.000000e+000	6.000000e+000	1.000000e+000	6.00000000
2	3.600000e+001	2.600000e+001	1.000000e+000	6.00000000
3	1.860000e+002	1.060000e+002	1.000000e+000	5.16666667
4	8.760000e+002	4.260000e+002	1.000000e+000	4.7096774
5	3.906000e+003	1.706000e+003	1.000000e+000	4.4589041
6	1.683600e+004	6.826000e+003	1.000000e+000	4.3102919
7	7.098600e+004	2.730600e+004	1.000000e+000	4.2163222
8	2.948760e+005	1.092260e+005	1.000000e+000	4.1540022
9	1.212306e+006	4.369060e+005	1.000000e+000	4.1112400

vlastní čísla: [4 3 1] $v_1 = [3, 1, 0]^T$, $v_2 = [1, 0, 0]^T$, $v_3 = [3, -2, 3]^T$

Poznámka:

$$y_0 = 1, \bar{6} \cdot v_1 - 5 \cdot v_2 + 0, \bar{3} \cdot v_3$$

V posloupnosti přibližných řešení vyšly dva po sobě jdoucí členy stejné.
Nejedná se ovšem o limitu posloupnosti přibližných řešení !

Posloupnost v komplexním oboru

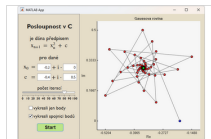
Opět budeme uvažovat iterační předpis, tentokrát v komplexním oboru.
Konkrétně se zaměříme na rekurentní předpis:

$$z_{n+1} = z_n^2 + c, \quad z_n, c \in \mathbb{C}$$

pro zadaná komplexní čísla z_0 a c .

O tom, zda iterační předpis konverguje či diverguje, rozhoduje konkrétní volba z_0 a c .

Následuje několik příkladů pro různé volby.



Posloupnost v komplexním oboru

by Josef Daněk

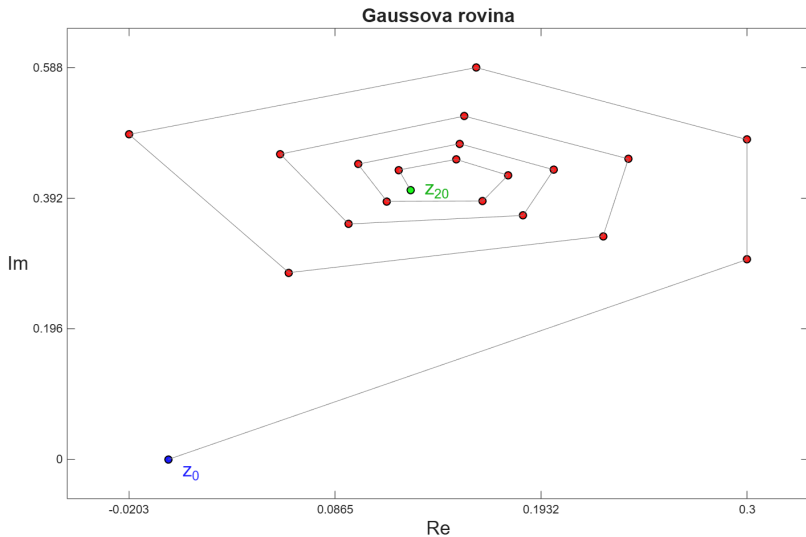
vykreslí N iterací zadané posloupnosti v komplexním oboru

version 1.0

pro testování jdi na <https://matlab.kma.zcu.cz>

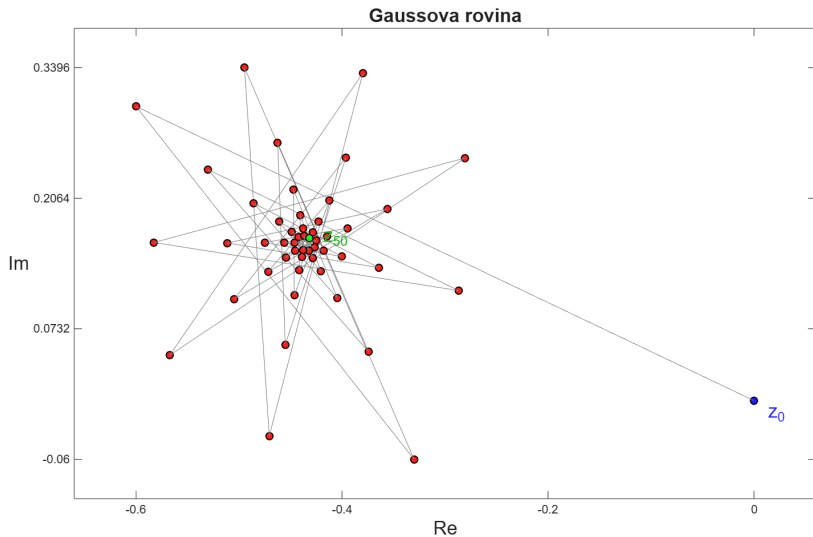
Posloupnost v komplexním oboru

$$z_0 = 0; \quad c = 0,3 + 0,3i$$



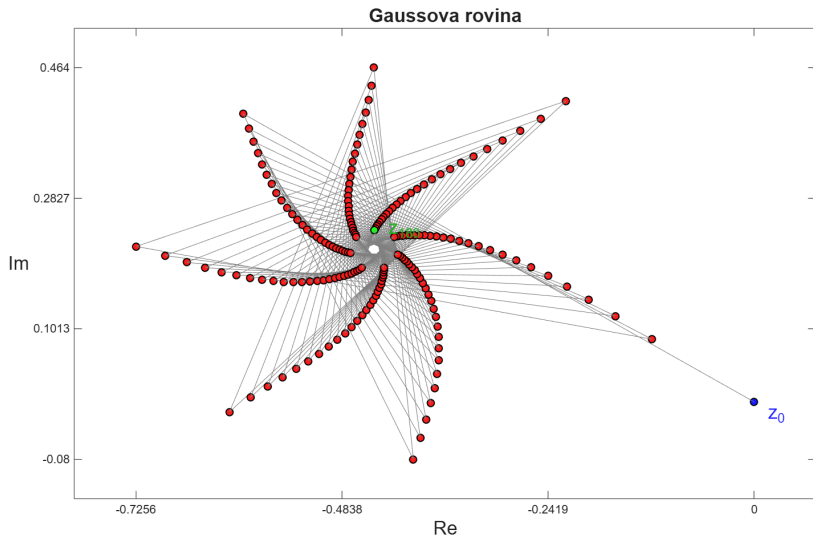
Posloupnost v komplexním oboru

$$z_0 = 0; \quad c = -0,6 + 0,3i$$



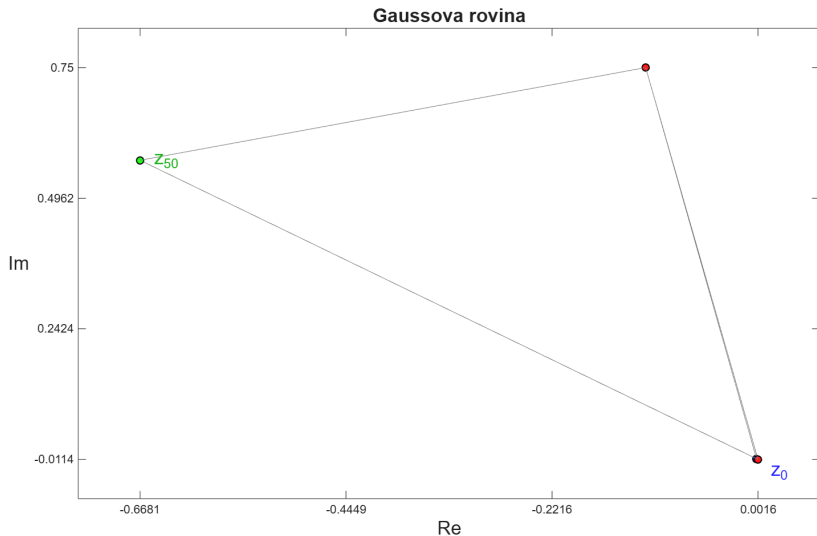
Posloupnost v komplexním oboru

$$z_0 = 0; \quad c = -0,6 + 0,4i$$



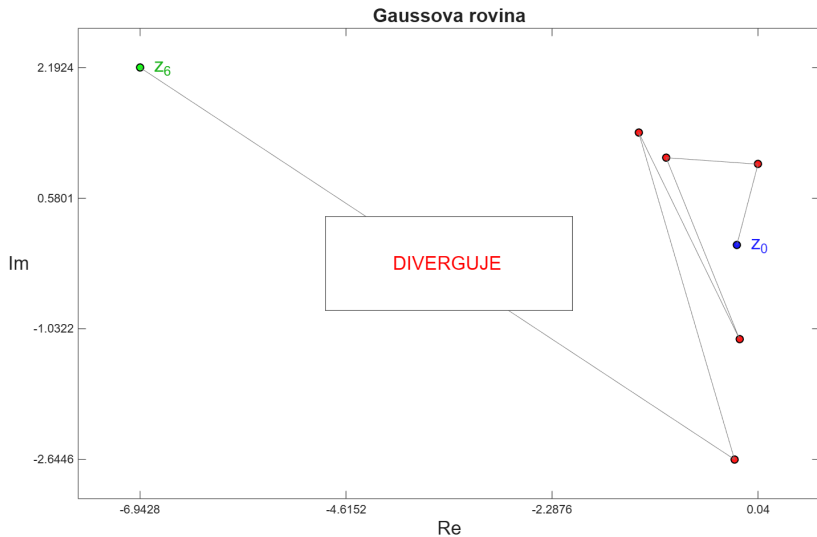
Posloupnost v komplexním oboru

$$z_0 = -0,01i; \quad c = -0,12 + 0,75i$$



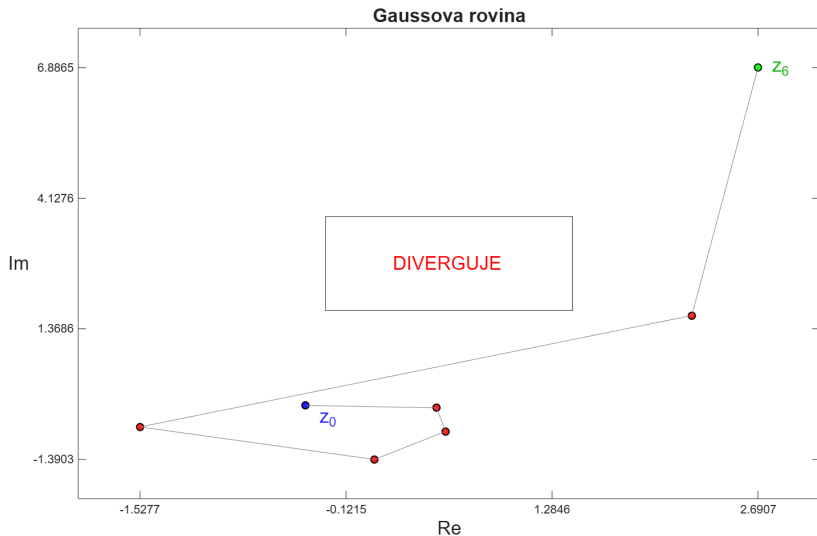
Posloupnost v komplexním oboru

$$z_0 = -0,2; \quad c = i$$



Posloupnost v komplexním oboru

$$z_0 = -0,4 - 0,25i; \quad c = 0,4 - 0,5i$$



Juliovy množiny

Stále uvažujeme iterační předpis v komplexním oboru:

$$z_{n+1} = z_n^2 + c, \quad z_n, c \in \mathbb{C}$$

z_n ... reprezentuje pozici bodu v komplexní rovině

c ... lib. zvolená konstanta, která je pro konkrétní obrazec stejná

$$\text{Juliova množina } J = \{z_0 \in \mathbb{C} : \lim_{n \rightarrow \infty} |z_n| \neq \infty\}$$

Při praktickém počítání máme k dispozici pouze konečný počet iterací, ze kterých musíme usoudit, zda posloupnost konverguje či nikoliv.

Juliovy množiny

Prahový poloměr divergence:

Pokud $\exists n \in \mathbb{N} : |z_n| > r(c) = \max\{|c|, 2\}$, potom posloupnost diverguje.

Důkaz:

$$|z_n| > 2 \text{ a } |z_n| > |c| \Rightarrow \exists \varepsilon > 0 : |z_n| = 2 + \varepsilon$$

Dále použijeme $\triangle$ nerovnost:

$$|z_n^2| = |z_n^2 + c - c| \leq |z_n^2 + c| + |c| \Rightarrow |z_n^2 + c| \geq |z_n^2| - |c|$$

Potom

$$\begin{aligned} |z_{n+1}| &= |z_n^2 + c| \geq |z_n^2| - |c| \geq |z_n|^2 - |c| \geq |z_n|^2 - |z_n| = \\ &= |z_n| \cdot (|z_n| - 1) > |z_n| \cdot (1 + \varepsilon) > 2 + 2\varepsilon \end{aligned}$$

V každé iteraci vzroste hodnota alespoň o faktor $(1 + \varepsilon)$,

tj. po k dalších krocích o faktor $(1 + \varepsilon)^k \Rightarrow z_n$ diverguje

Vykreslení

- určíme oblast v komplexní rovině
(vzhledem k prahovému poloměru konvergence)

$$-2 \leq \operatorname{Re} z \leq 2, \quad -2 \leq \operatorname{Im} z \leq 2$$

- zvolíme konstantu c a diskretizační krok
- postupně pro každý bod z_0 počítáme podle iterační formule

$$z_{n+1} = z_n^2 + c$$

- jestliže $|z_n|$ překročí hodnotu $\max\{|c|, 2\}$ řekneme, že posloupnost diverguje
- pokud po zadaném počtu iterací nepřekročí $|z_n|$ hodnotu $\max\{|c|, 2\}$, rozhodneme, že posloupnost konverguje
- body v komplexní rovině obarvíme podle toho zda posloupnost konvergovala či divergovala

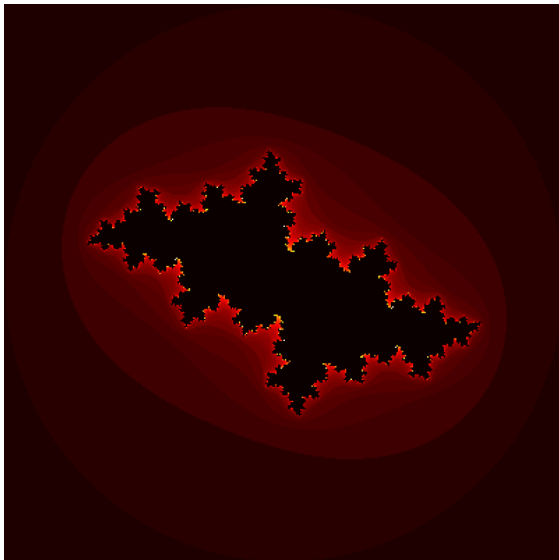
Juliovy množiny

$$c = -0,097 - 0,64872i$$



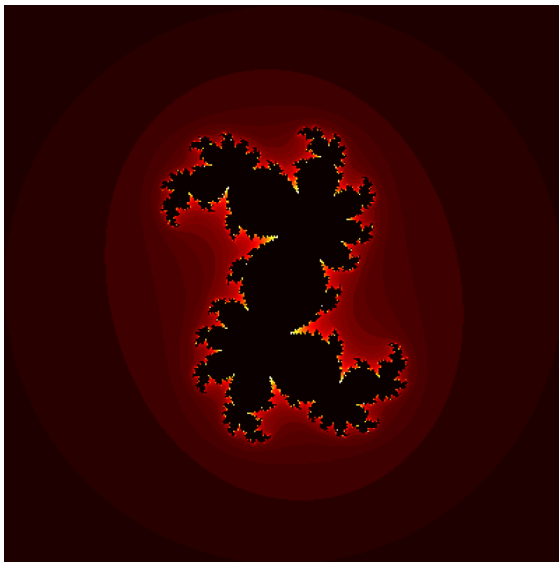
Juliovy množiny

$$c = -0,504 + 0,50197i$$



Juliovy množiny

$$c = 0,37688 + 0,21646i$$



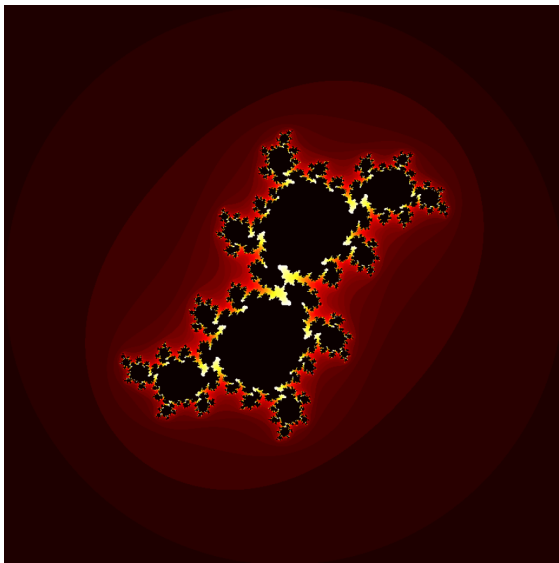
Juliovy množiny

$$c = -0,3555 - 0,62008i$$



Juliovy množiny

$$c = 0,060124 - 0,62423i$$



Juliovy množiny

$$c = -0,7515 - 0,072i$$



Mandelbrotova množina

Opět i zde použijeme iterační předpis:

$$z_{n+1} = z_n^2 + c, \quad z_n, c \in \mathbb{C}$$

ovšem zde volíme počáteční hodnotu $z_0 = 0$

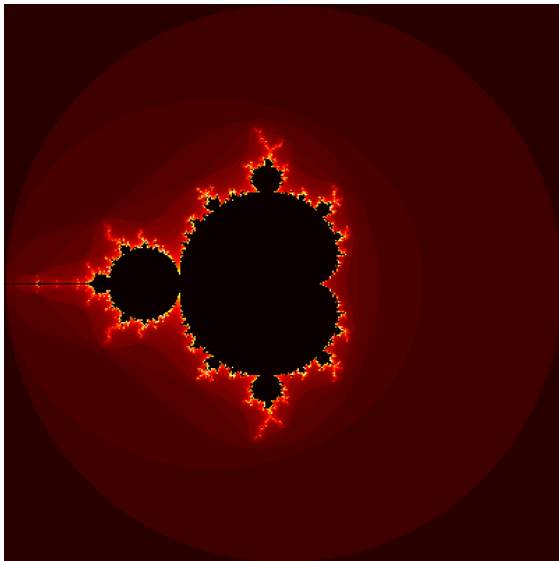
c ... reprezentuje pozici bodu v komplexní rovině

$$\text{Mandelbrotova množina} \quad M = \{c \in \mathbb{C} : \lim_{n \rightarrow \infty} |z_n| \neq \infty\}$$

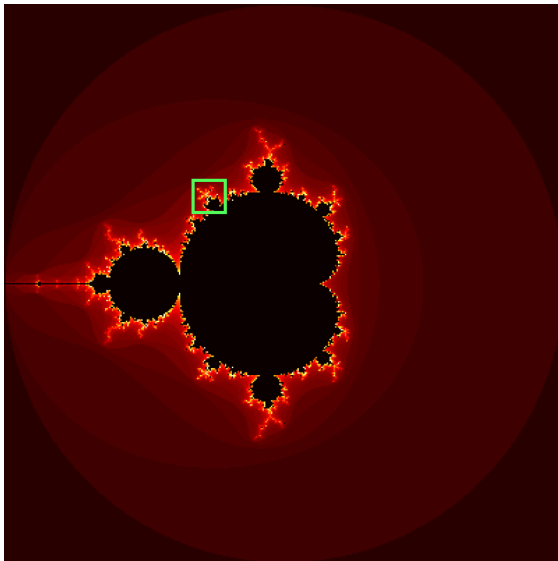
Mandelbrotova množina je pouze jedna, na rozdíl od Juliových množin, kde pro každou volbu konstanty $c \in \mathbb{C}$ dostaneme jinou množinu.

Vykreslíme znovu na oblasti $-2 \leq \operatorname{Re} z \leq 2$, $-2 \leq \operatorname{Im} z \leq 2$

Mandelbrotova množina

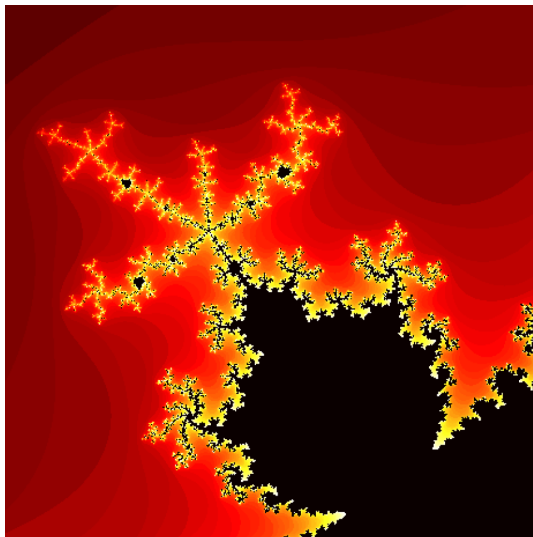


Mandelbrotova množina



Mandelbrotova množina

Detail na oblasti $-0,65 \leq \operatorname{Re} z \leq -0,42$, $0,51 \leq \operatorname{Im} z \leq 0,74$



Deterministický chaos (χαος)

Deterministický chaos - na první pohled slova s protichůdným obsahem

Edward Norton Lorenz (23.5.1917 - 16.4.2008) (*otec chaosu*)

Americký matematik (Harvard) a meteorolog (MIT). Při studiu modelů počasí objevil, že počasí se ne vždy chová podle předpovědi. Malá odchylka v počátečních hodnotách v jeho počítačovém modelu počasí měla za následek velké rozdíly v chování počasí. Tato citlivá závislost na počátečních podmínkách se stala známou jako motýlí efekt.

- **Vylučuje prediktabilitu** (dlouhodobější)
- **Deterministické chování** - ve zdánlivě chaotických systémech (turbulence, stoupající kour) existuje skrytý řád. Je možné stanovit, jaké příčiny vedly k danému pozorovanému následku (většinou je to však možné pouze dodatečně - *a posteriori*)
- **Soběpodobnost** - určitá vlastnost se zachovává nezávisle na měřítku, bjevuje se určitý řád ve zdánlivém chaosu; graficky znázorněná vlastnost soběpodobnosti ... fraktál
- **Samoorganizace** - například pohyb ryb v hejnu; každá ryba se řídí třemi jednoduchými lokálními pravidly: soudržnost hejna, zařazení a oddělení

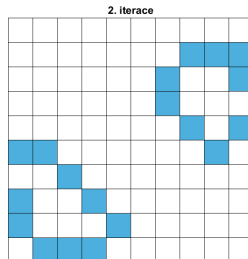
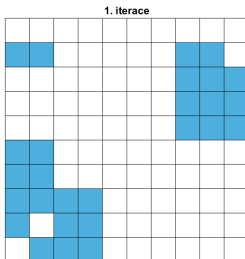
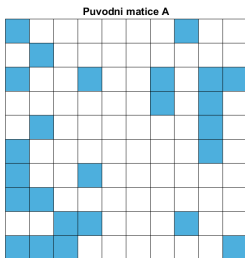
Je překvapující, jak komplikované kolektivní chování z těchto 3 jednoduchých pravidel vychází. Neexistuje žádný globální plán pohybu nebo perspektiva a neexistuje žádný vůdce hejna (podobné chování mravenců).

Deterministický chaos

Hra život

pro testování jdi na
<https://matlab.kma.zcu.cz>

```
function zivot(A,p,q,N);  
% trivialni podoba hry ZIVOT  
% A ... zadana binarni matice  
%     1 ... ziva bunka  
%     0 ... mrtva bunka  
% okoli bunky [i,j]:  
%     [i+1,j-1] [i+1, j ] [i+1,j+1]  
%     [ i ,j-1] [ i , j ] [ i ,j+1]  
%     [i-1,j-1] [i-1, j ] [i-1,j+1]  
% p ... min pocet bunek v okoli pro zachovani zivota  
% q ... max pocet bunek v okoli pro zachovani zivota  
% N ... max pocet iteraci
```

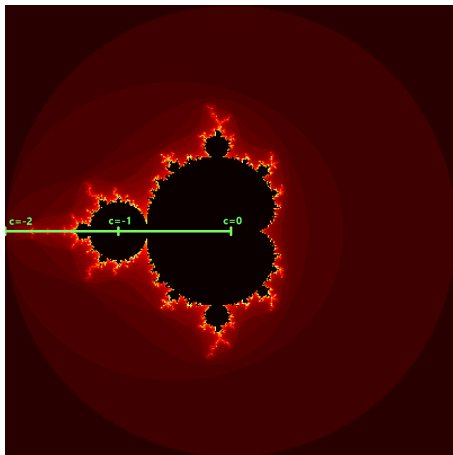


Deterministický chaos

Opět uvažujeme iterační předpis použitý u Mandelbrotovy množiny:

omezíme se na reálný obor $\mathbb{R}$: $z_{n+1} = z_n^2 + c, \quad z_n, c \in \mathbb{R}$

a dále volíme počáteční hodnotu $z_0 = 0$ a $c \in \langle -2; 0 \rangle$

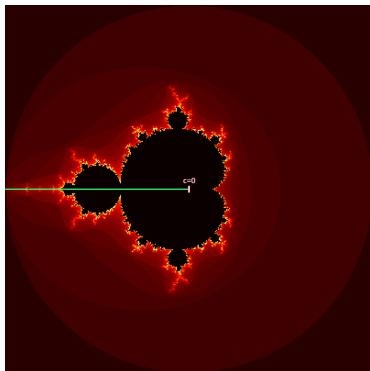


Deterministický chaos

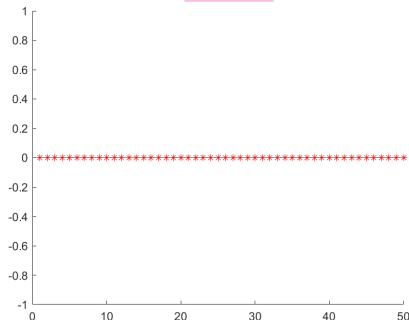
$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotovy množiny



$c = 0$

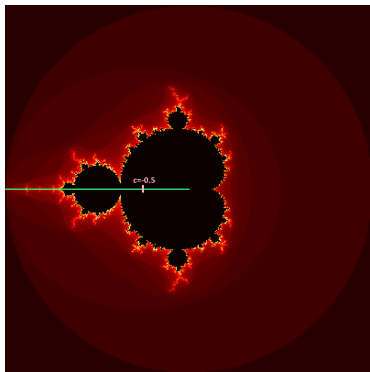


Deterministický chaos

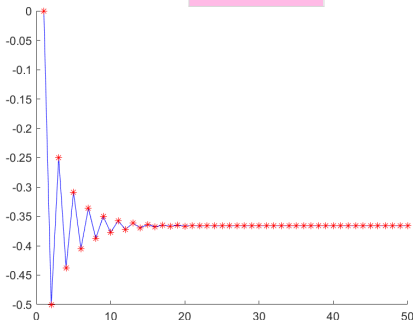
$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotové množiny



$$c = -0,5$$

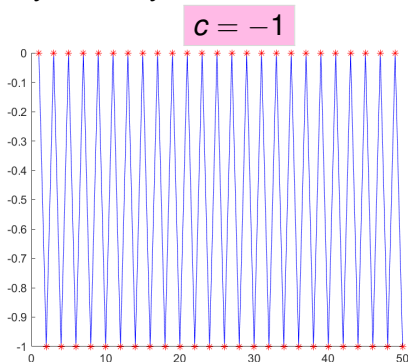
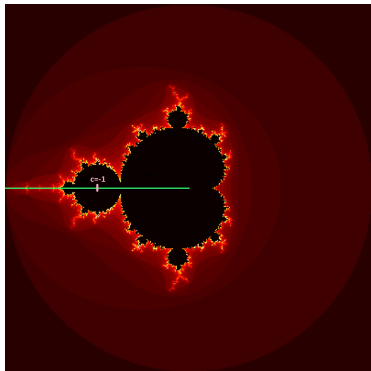


Deterministický chaos

$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotovy množiny

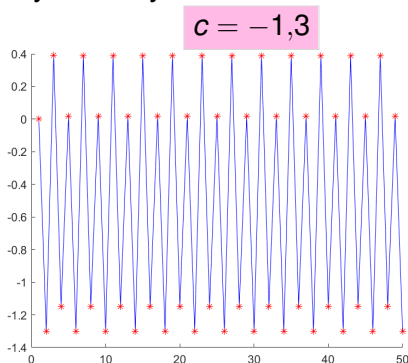
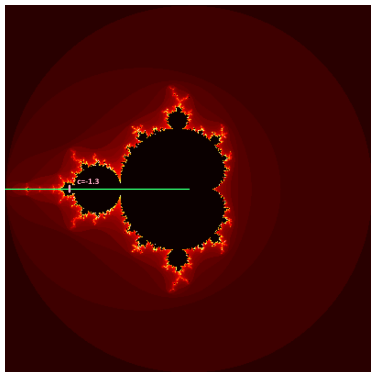


Deterministický chaos

$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotové množiny

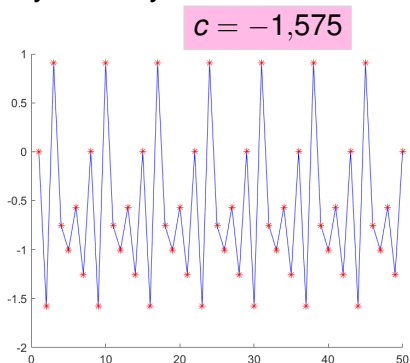
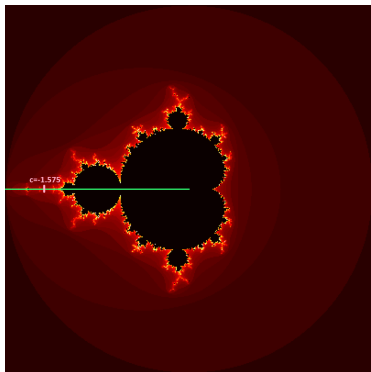


Deterministický chaos

$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotové množiny

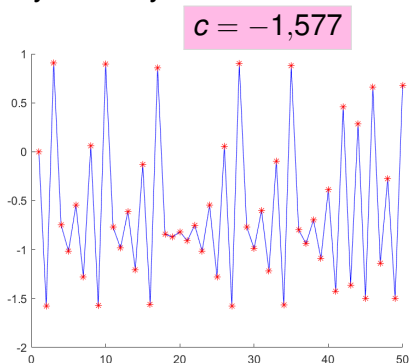
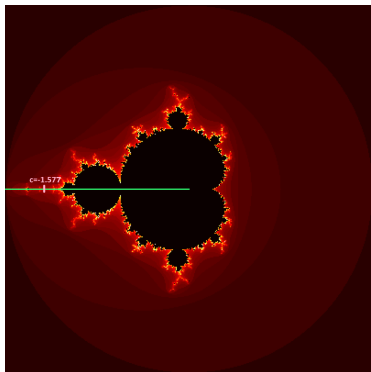


Deterministický chaos

$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotové množiny

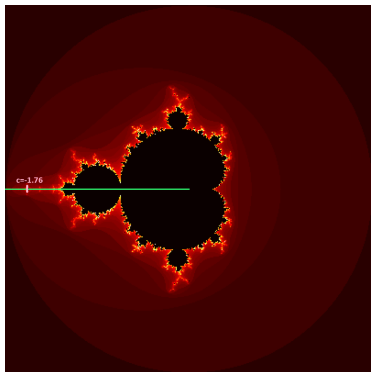


Deterministický chaos

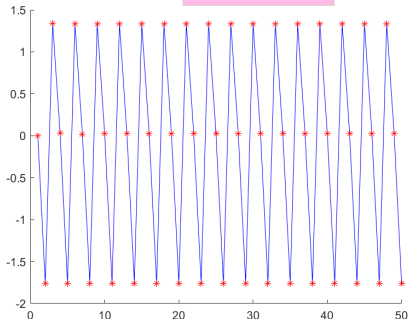
$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotové množiny



$$c = -1,76$$

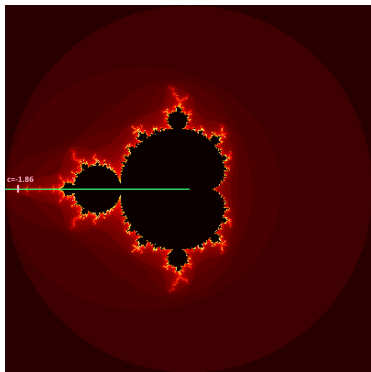


Deterministický chaos

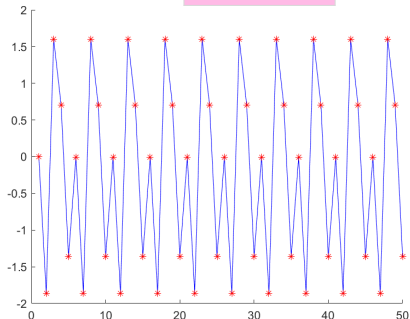
$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotové množiny



$$c = -1,86$$

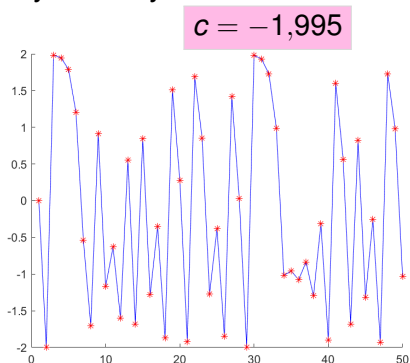
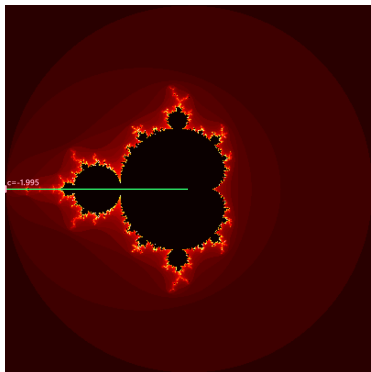


Deterministický chaos

$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotové množiny

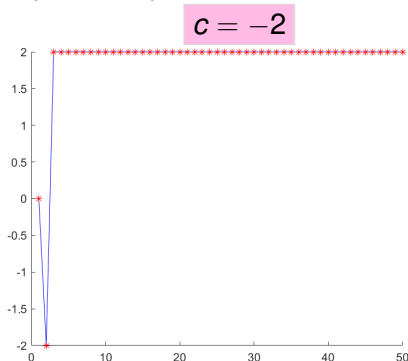
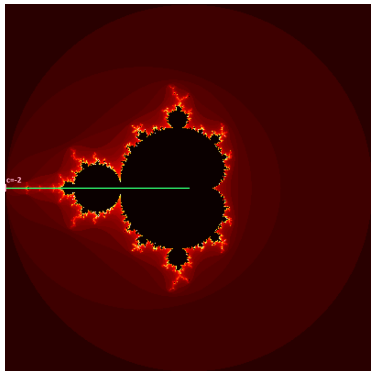


Deterministický chaos

$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotovy množiny

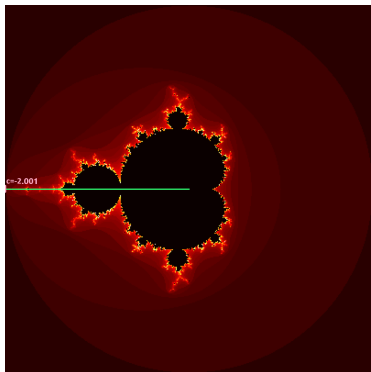


Deterministický chaos

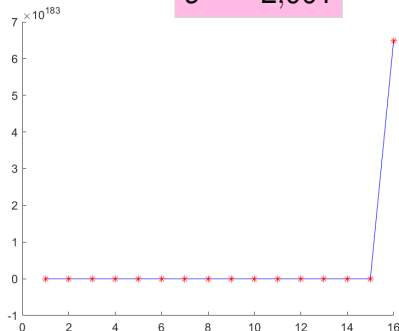
$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

Sledujme posloupnost iterací pro několik různých voleb konstanty c :

- pro $c \in \langle -2; 0 \rangle$ posloupnost nedivergovala
- tj. tyto body patřily do Mandelbrotové množiny

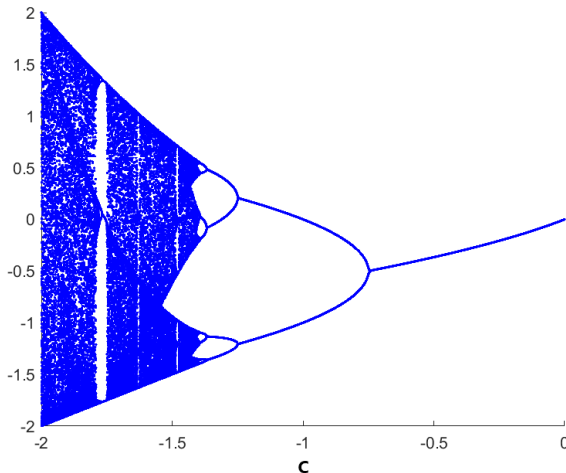


$$c = -2,001$$

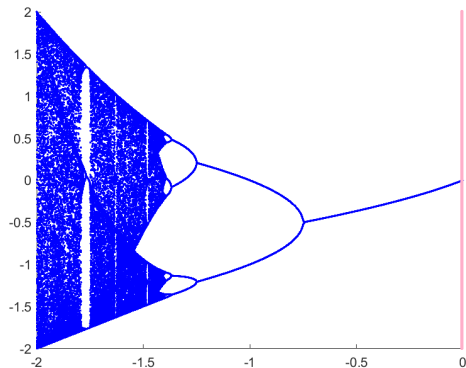


Deterministický chaos

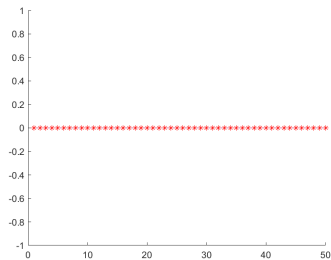
- pro $c \in \langle -2; 0 \rangle$ vypočteme prvních 1000 iterací dané posloupnosti
- pro každou hodnotu c vykreslíme do grafu posledních 50 iterací
- tj. pokud byly hodnoty stejné, vykreslí se jako jeden bod



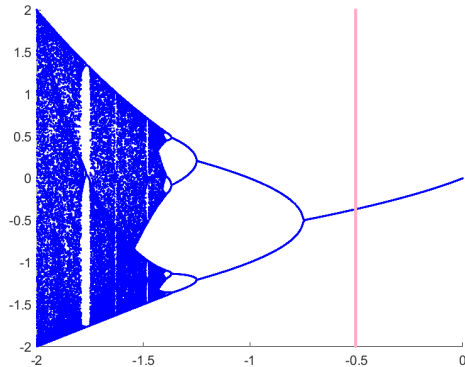
Deterministický chaos



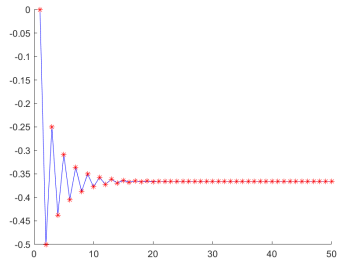
$$c = 0$$



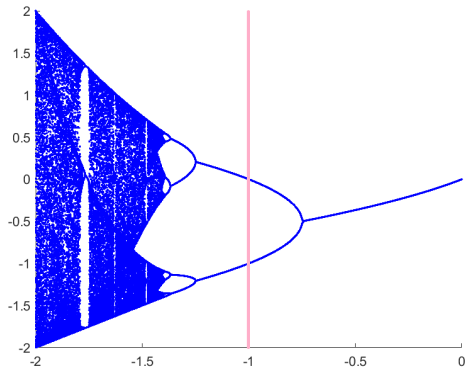
Deterministický chaos



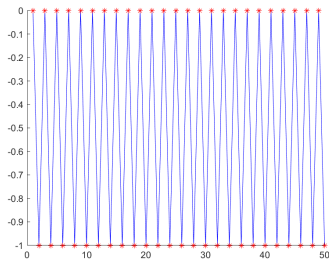
$$c = -0,5$$



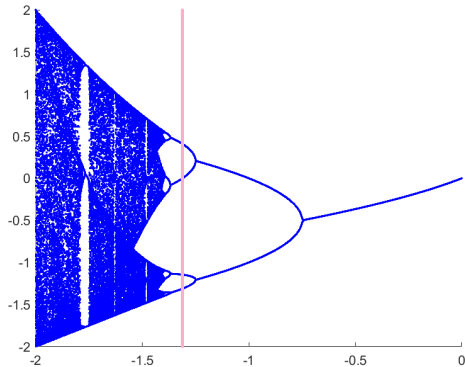
Deterministický chaos



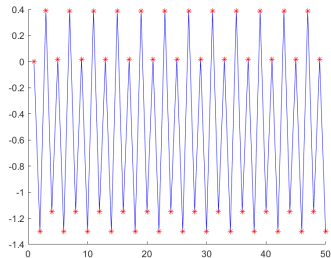
$$c = -1$$



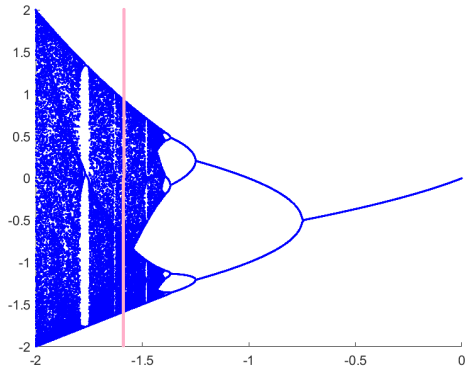
Deterministický chaos



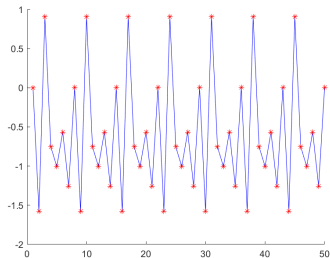
$$c = -1,3$$



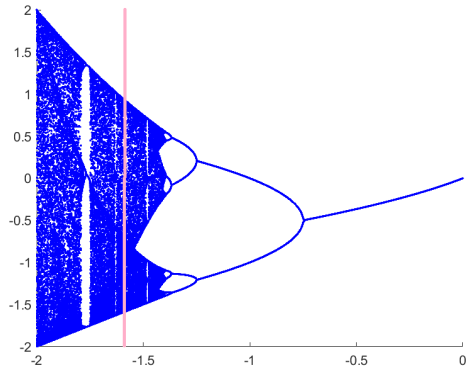
Deterministický chaos



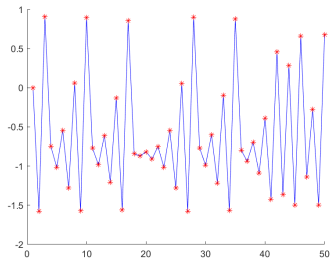
$$c = -1,575$$



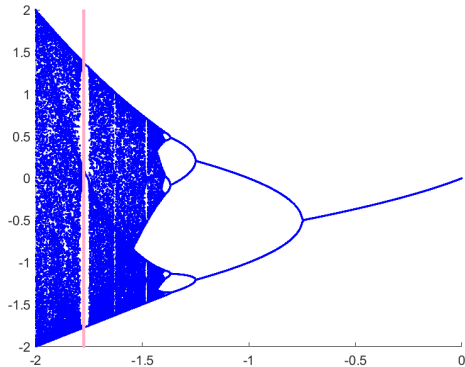
Deterministický chaos



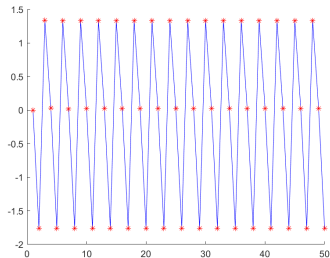
$$c = -1,577$$



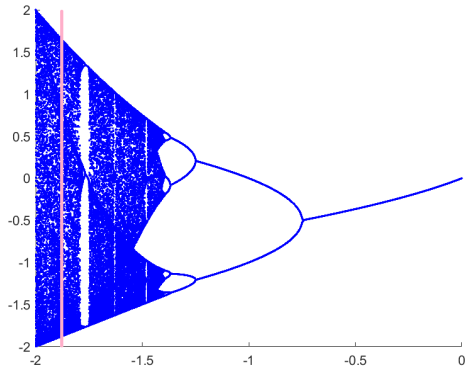
Deterministický chaos



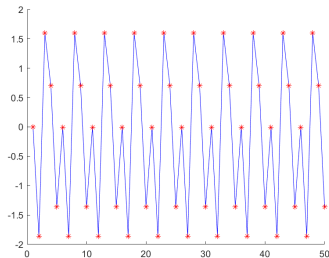
$$c = -1,76$$



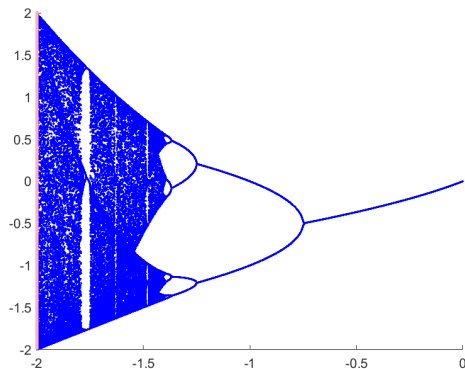
Deterministický chaos



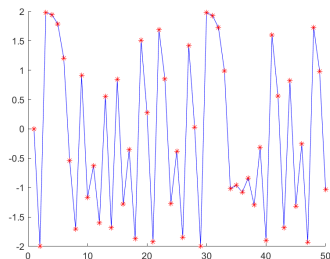
$$c = -1,86$$



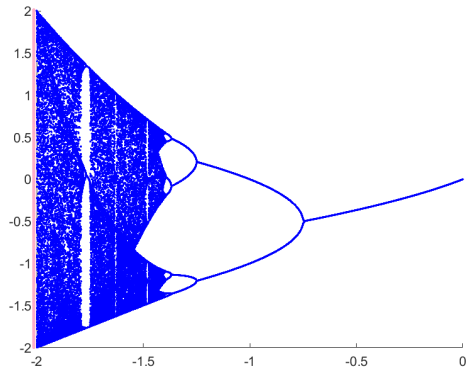
Deterministický chaos



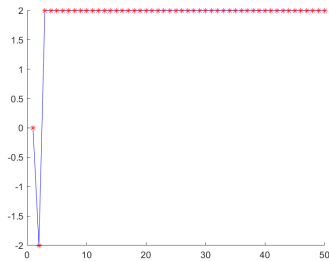
$$c = -1,995$$



Deterministický chaos



$$c = -2$$

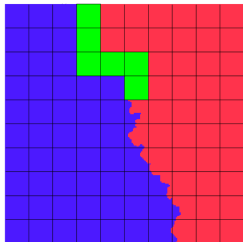


Měření délky pobřeží

Jak lze změřit délku pobřeží?

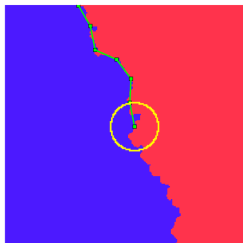
Pomocí mřížky:

- vytvoříme mřížku (čtvercovou) s určitou velikostí čtverce
- obarvíme ty čtverce, ve kterých leží jak pevnina, tak moře
- délka je dána počtem obarvených čtverců $\times$ jejich počet (strana, úhlopříčka)



Pomocí kružítka:

- nastavíme délku měřidla (kružítka)
- začneme v jednom krajním bodě
- určíme další bod pobřeží, který leží právě tak daleko od předchozího
- délka je součtem vzdáleností získaných bodů, tj. jejich počet $\times$ délka měřidla

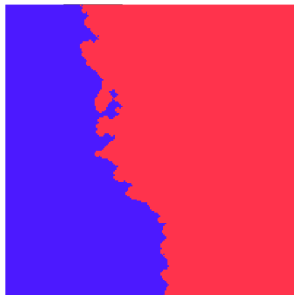


Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

Měření délky pobřeží pomocí čtvercové sítě
necht mapa představuje území 100 x 100 km



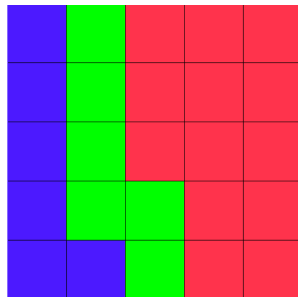
Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

Merění délky pobřeží pomocí čtvercové sítě
pokud mapa představuje území 100 x 100 km
zvolený čtverec má stranu délky 20.000 km
počet označených čtverců je 6
celková délka je $6 \times 20.000 = 120.000$ km
(součin počtu čtverců a délky jeho strany)

120,000 km



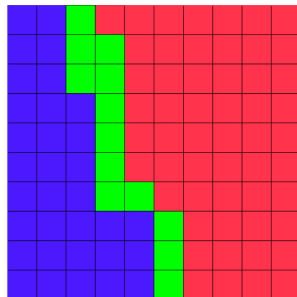
Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

Mereni delky pobrezi pomoci ctvercove site
pokud mapa predstavuje uzemi 100 x 100 km
zvoleny ctverec ma stranu delky 10.000 km
pocet oznacenych ctvercu je 13
celkova delka je $13 \times 10.000 = 130.000$ km
(sucin poctu ctvercu a delky jeho strany)

130,000 km



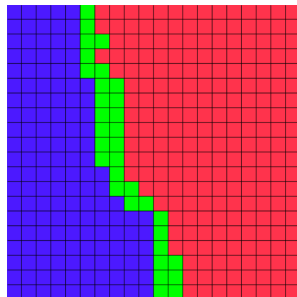
Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

Merění délky pobřeží pomocí čtvercové sítě
pokud mapa představuje území 100×100 km
zvolený čtverec má stranu délky 5.000 km
počet označených čtverců je 33
celková délka je $33 \times 5.000 = 165.000$ km
(součin počtu čtverců a délky jeho strany)

165,000 km



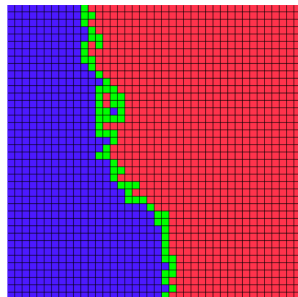
Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

Mereni delky pobrezi pomoci ctvercove site
pokud mapa predstavuje uzemi 100 x 100 km
zvoleny ctverec ma stranu delky 2.500 km
pocet oznacenych ctvercu je 67
celkova delka je $67 \times 2.500 = 167.500$ km
(sucin poctu ctvercu a delky jeho strany)

167,500 km

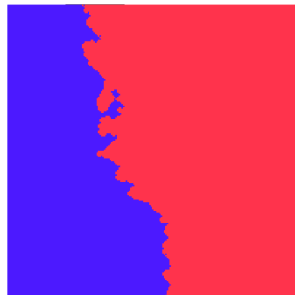


Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

Měření délky pobřeží pomocí kruzítka
necht mapa představuje území 100 x 100 km



Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

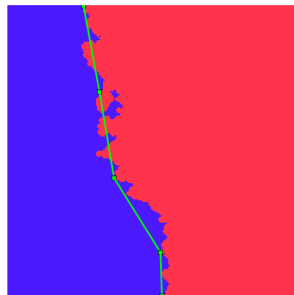
Merění délky pobřeží pomocí kruzítka

necht mapa představuje území 100 x 100 km
nastavená délka v kruzítku je 30 km

pocet naměřených úsečků je 3.5003

celková délka je $3.5003 \times 30 = 105.009$ km
(součin počtu úsečků a její délky)

105,009 km



Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

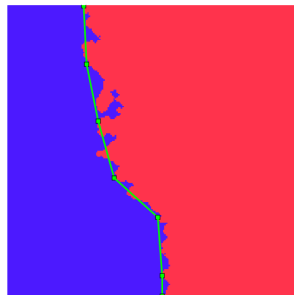
Měření délky pobřeží pomocí kruzítka

necht mapa představuje území 100×100 km
nastavená délka v kruzítku je 20 km

počet naměřených úsečků je 5.3500

celková délka je $5.3500 \times 20 = 107.000$ km
(součin počtu úsečků a její délky)

107,000 km



Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

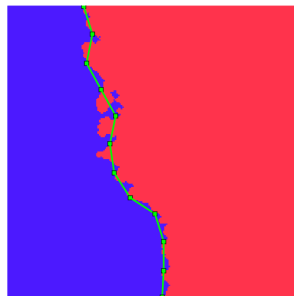
Merění délky pobřeží pomocí kruzítka

necht mapa představuje území 100×100 km
nastavená délka v kruzítku je 10 km

počet naměřených úsečků je 10.9014

celková délka je $10.9014 \times 10 = 109.014$ km
(součin počtu úsečků a její délky)

109,014 km



Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

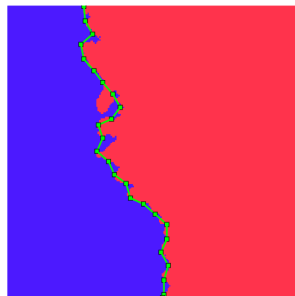
Merění délky pobřeží pomocí kruzítka

necht mapa představuje území 100 x 100 km
nastavená délka v kruzítku je 5 km

pocet naměřených úseček je 24.2236

celková délka je $24.2236 \times 5 = 121.118$ km
(součin počtu úseček a její délky)

121,118 km



Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

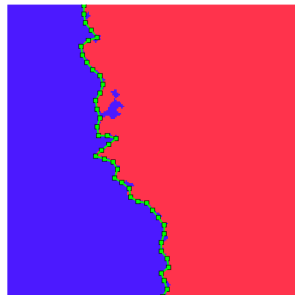
Merění délky pobřeží pomocí kruzítka

necht mapa představuje území 100 x 100 km
nastavená délka v kruzítku je 3 km

pocet naměřených úsečků je 42.9718

celková délka je $42.9718 \times 3 = 128.915$ km
(součin počtu úsečků a její délky)

128,915 km



Měření délky pobřeží

Richardsonův efekt

Čím menší budu mít měřidlo tím větší naměříme délku pobřeží !
Členitost pobřeží charakterizuje neceločíselná dimenze.

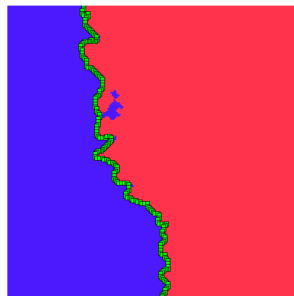
Merění délky pobřeží pomocí kruzítka

necht mapa představuje území 100×100 km
nastavená délka v kruzítku je 1.5 km

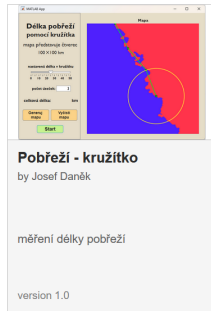
pocet naměřených úsečků je 186.6667

celková délka je $186.6667 \times 1.5 = 280$ km
(součin počtu úsečků a její délky)

280,000 km



Měření délky pobřeží



pro testování jdi na <https://matlab.kma.zcu.cz>

Kochova křivka

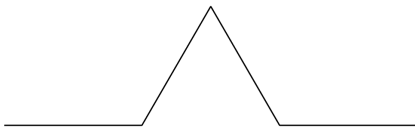
Uvažujeme následující fraktálovou strukturu:

0) nejprve uvažujeme úsečku délky jedna



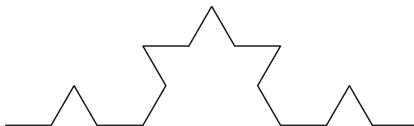
Kochova křivka

- 1) prostřední třetinu úsečky nahradíme dvěma stejně velkými úsečkami, takže s původní částí tvoří rovnostranný trojúhelník



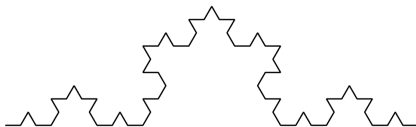
Kochova křivka

- 2) nad každou stranou této lomené čáry provedeme stejnou operaci a neustále opakujeme



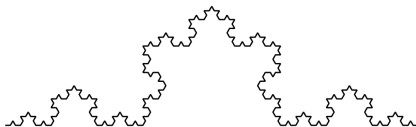
Kochova křivka

- 3) nad každou stranou této lomené čáry provedeme stejnou operaci a neustále opakujeme



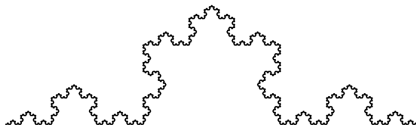
Kochova křivka

- 4) nad každou stranou této lomené čáry provedeme stejnou operaci a neustále opakujeme



Kochova křivka

- 5) nad každou stranou této lomené čáry provedeme stejnou operaci a neustále opakujeme



Kochova křivka

Otázka: Jak dlouhá je lomená čára ?

Pro délku v jednotlivých krocích dostáváme:

- 0) Na počátku: 1 úsečka o délce jedna
- 1) Po 1. kroku: 4 úsečky o délce $1/3$
- 2) Po 2. kroku: 16 úseček o délce $1/9$
- 3) Po 3. kroku: 64 úseček o délce $1/27$
- $\vdots$
- n) Po n krocích: 4^n úseček o délce $1/3^n$

Limitní přechod:

celková délka lomené čáry je $\lim_{n \rightarrow \infty} 4^n \cdot \left(\frac{1}{3}\right)^n = \lim_{n \rightarrow \infty} \left(\frac{4}{3}\right)^n = \infty$

Kochova křivka

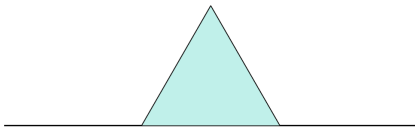
Jak velký obsah vymezí Kochova křivka nad původní úsečkou:

0) nejprve uvažujeme úsečku délky jedna



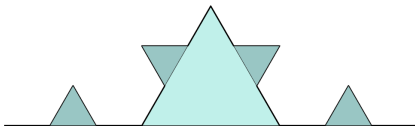
Kochova křivka

- 1) prostřední třetinu úsečky nahradíme dvěma stejně velkými úsečkami, takže s původní částí tvoří rovnostranný trojúhelník



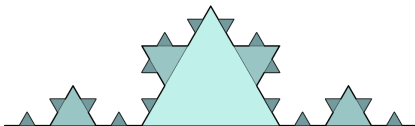
Kochova křivka

- 2) nad každou stranou této lomené čáry provedeme stejnou operaci a neustále opakujeme



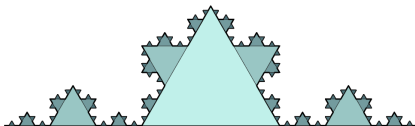
Kochova křivka

- 3) nad každou stranou této lomené čáry provedeme stejnou operaci a neustále opakujeme



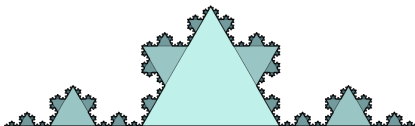
Kochova křivka

- 4) nad každou stranou této lomené čáry provedeme stejnou operaci a neustále opakujeme



Kochova křivka

- 5) nad každou stranou této lomené čáry provedeme stejnou operaci a neustále opakujeme



Kochova křivka

Otázka: Jaký obsah vymezí lomená čára nad úsečkou ?

Pro obsah plochy v jednotlivých krocích dostáváme:

- 0) Na počátku žádný obsah
- 1) Po 1. kroku přibude 1 trojúhelník o straně $1/3$
- 2) Po 2. kroku přibude 4 trojúhelníky o straně $1/9$
- 3) Po 3. kroku přibude 16 trojúhelníků o straně $1/27$
- $\vdots$
- n) Po n krocích přibude 4^n trojúhelníků o straně délky $1/3^n$

obsah rovnostranného trojúhelníka o straně a je

$$S_{\triangle} = \frac{1}{2}a \cdot v = \frac{1}{2}a \cdot \frac{\sqrt{3}}{2}a = \frac{\sqrt{3}}{4}a^2$$

Kochova křivka

Otázka: Jaký obsah vymezí lomená čára nad úsečkou ?

Limitní přechod pro celkový obsah plochy pod lomenou čarou:

$$\sum_{n=1}^{\infty} 4^{n-1} \cdot \frac{\sqrt{3}}{4} \cdot \left(\frac{1}{3}\right)^{2n} = \sum_{n=1}^{\infty} \frac{\sqrt{3}}{16} \cdot 4^n \cdot \left(\frac{1}{9}\right)^n = \sum_{n=1}^{\infty} \frac{\sqrt{3}}{16} \cdot \left(\frac{4}{9}\right)^n =$$

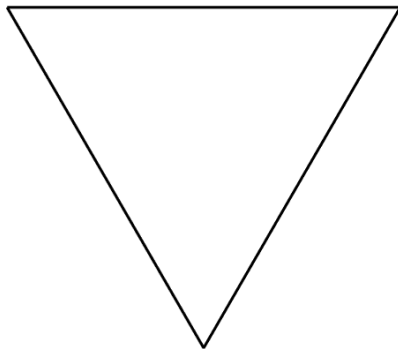
součet nekonečné geometrické řady s kvocientem $q = \frac{4}{9}$ a prvním členem $\frac{\sqrt{3}}{36}$

$$= \frac{\sqrt{3}}{36} \cdot \frac{1}{1 - \frac{4}{9}} = \frac{\sqrt{3}}{36} \cdot \frac{9}{5} = \frac{\sqrt{3}}{20} \doteq 0,08660254.$$

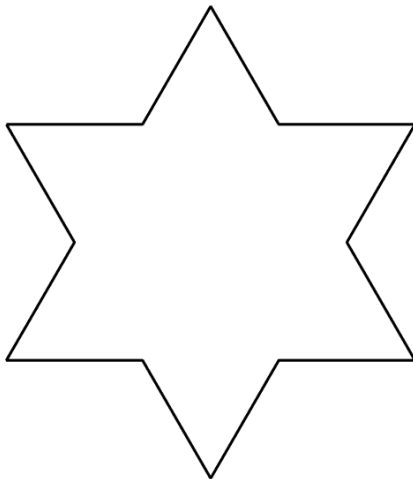
Závěr: Objekt má konečný obsah, ale nekonečný obvod !

Poznámka: Podobně je tomu v případě měření délky pobřeží.

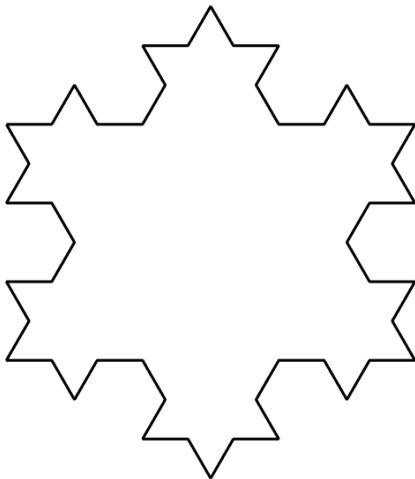
Kochova vložka



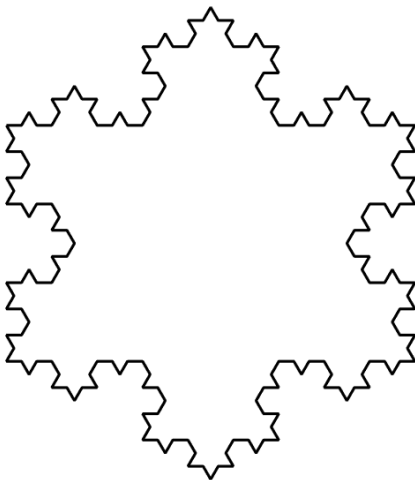
Kochova vložka



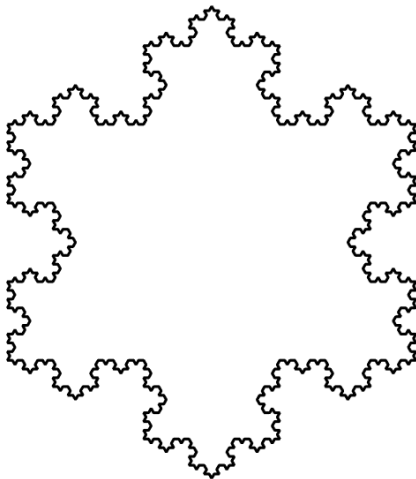
Kochova vločka



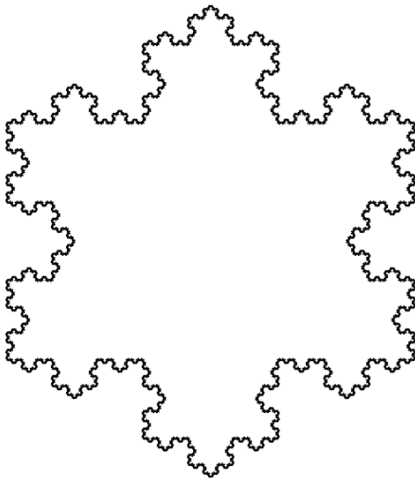
Kochova vložka



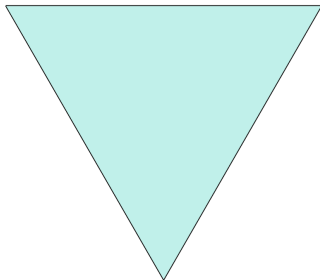
Kochova vložka



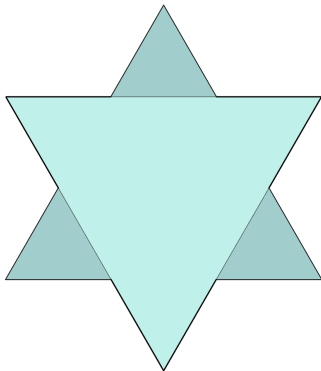
Kochova vločka



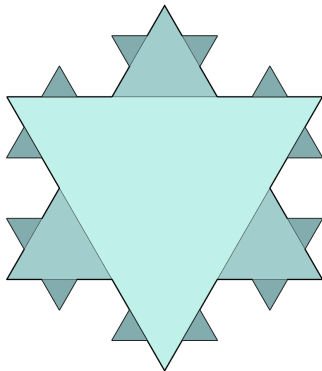
Kochova vložka



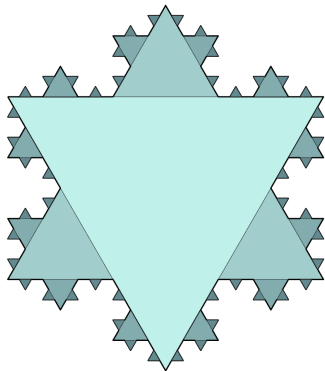
Kochova vločka



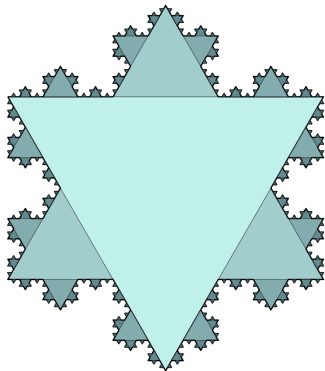
Kochova vložka



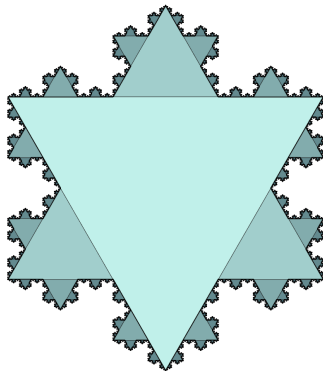
Kochova vločka



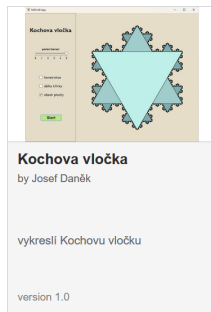
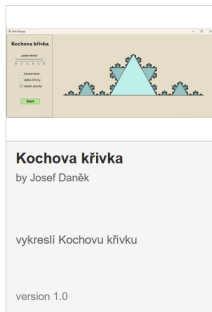
Kochova vločka



Kochova vločka



Kochova vločka



pro testování jdi na <https://matlab.kma.zcu.cz>

Fraktální dimenze křivky

- uvažujeme měření délky pomocí měřidla délky r
- aproximace délky křivky je dána počtem kroků měřidla podél křivky (N) vynásobené délkou měřidla $E(r) = N \cdot r$
- pro hladké křivky konverguje $E(r)$ pro $r \rightarrow 0_+$ ke konečnému číslu ... délce křivky ... dimenze je 1
(stačí popsat pouze vzdálenost od nějakého pevného bodu)
- pro fraktály $E(r)$ diverguje k $+\infty \Rightarrow$ dimenze je větší než 1
- chceme najít hodnotu D_H takovou, že

$N \cdot r^{D_H}$ je konečné nenulové číslo

D_H ... fraktální dimenze křivky (Hausdorfova dimenze)

Fraktální dimenze křivky

Pro $r \rightarrow 0_+$ a $N \rightarrow \infty$ má platit ($0 < c < \infty$):

$$N \cdot r^{D_H} = c \quad / \log$$

$$\log(N \cdot r^{D_H}) = \log c$$

$$\log N + D_H \log r = \log c$$

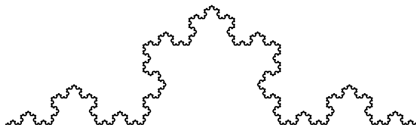
$$D_H = \frac{\log c - \log N}{\log r}$$

$$D_H = \frac{\log N - \log c}{\log \frac{1}{r}} = \frac{\log N}{\log \frac{1}{r}} + \frac{\log c}{\log r} \quad \left(\frac{\log c}{\log r} \rightarrow 0 \text{ pro } r \rightarrow 0_+ \right)$$

$$D_H = \lim_{r \rightarrow 0_+, N \rightarrow \infty} \frac{\log N}{\log \frac{1}{r}}$$

Fraktální dimenze křivky

pro **Kochovu křivku**:



v n -tém kroku jsme měli $\underbrace{4^n}_{=N}$ úseček o délce $\underbrace{\left(\frac{1}{3}\right)^n}_{=r}$

$$D_H = \lim_{N \rightarrow \infty, r \rightarrow 0+} \frac{\log N}{\log \frac{1}{r}} = \lim_{n \rightarrow \infty} \frac{\log 4^n}{\log 3^n} = \frac{\log 4}{\log 3} \doteq \underline{1,2619}$$

(neceločíselná hodnota dimenze)

Fraktální dimenze křivky

Příklad jiné fraktálové struktury - **Cantorovo diskontinuum**:

0) nejprve uvažujeme úsečku délky jedna



1) prostřední třetinu úsečky vypustíme



2) nad každou úsečkou provedeme stejnou operaci



3) nad každou úsečkou provedeme stejnou operaci



4) nad každou úsečkou provedeme stejnou operaci



5) nad každou úsečkou provedeme stejnou operaci



Fraktální dimenze křivky

pro **Cantorovo diskontinuum**:

v n -tém kroku jsme měli $\underbrace{2^n}_{=N}$ úseček o délce $\underbrace{\left(\frac{1}{3}\right)^n}_{=r}$

celková délka lomené čáry je $\lim_{n \rightarrow \infty} 2^n \cdot \left(\frac{1}{3}\right)^n = \lim_{n \rightarrow \infty} \left(\frac{2}{3}\right)^n = 0$

$$D_H = \lim_{N \rightarrow \infty, r \rightarrow 0_+} \frac{\log N}{\log \frac{1}{r}} = \lim_{n \rightarrow \infty} \frac{\log 2^n}{\log 3^n} = \frac{\log 2}{\log 3} \doteq \underline{0,6309}$$

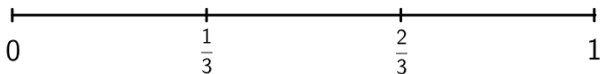
Fraktální dimenze křivky

pro **Cantorovo diskontinuum** jsme naznačili 5 iterací (aproximací)

- množina samotná je limitním případem pro počet iterací $\rightarrow \infty$
- jak si ji máme představit?
- na pomyslném nekonečném řádku je jen „roztroušený prach“
- Cantorovo diskontinuum neobsahuje žádný souvislý interval
- pro zápis čísel použijme trojkovou soustavu (cifry 0, 1 a 2)

čísla z intervalu $(0;1)$ v trojkové soustavě $(0, \dots\dots\dots)_3$

$$0 = (0,0)_3 = 0 \cdot 3^{-1} \quad \frac{1}{3} = (0,1)_3 = 1 \cdot 3^{-1} \quad \frac{2}{3} = (0,2)_3 = 2 \cdot 3^{-1}$$



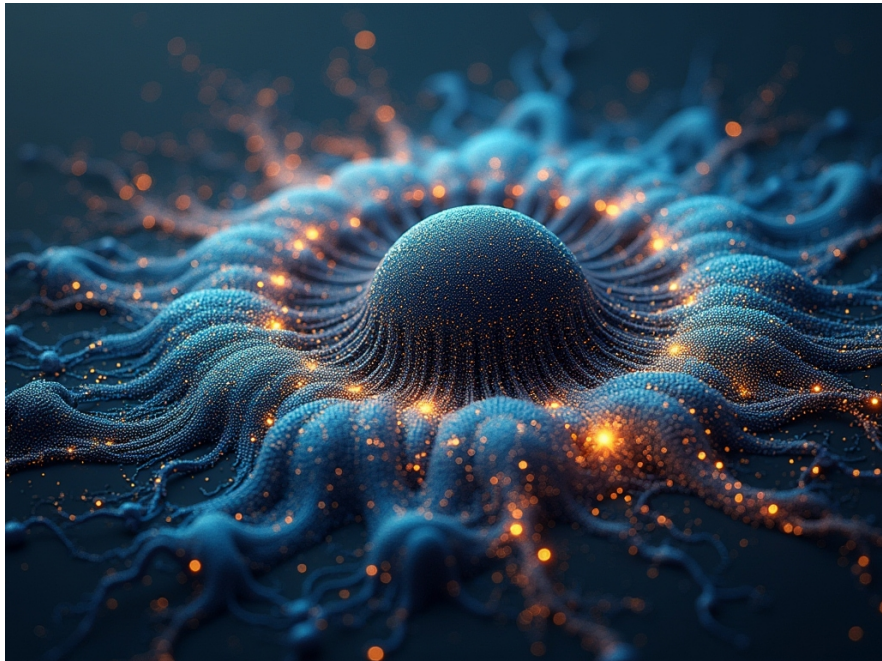
- v 1. iteraci jsme vyhodili prostřední třetinu, tj. čísla, jejichž první číslice v trojkové soustavě za čárkou je 1

Fraktální dimenze křivky

- v 2. iteraci jsme ze zbývajících třetin vyhodili prostřední třetinu, tj. všechna čísla, jejichž druhá číslice (v trojkové soustavě) je 1
- v n -té iteraci z jednotkového intervalu vyhadujeme čísla, jejichž n -tá číslice je 1
- iterační postup tedy postupně požírá čísla, jejichž zápis v trojkové soustavě obsahuje někde číslici 1
- zůstanou čísla, jejichž trojkový zápis neobsahuje žádnou jedničku např. $(0,200202202222020002)_3$
(a takové snadno lze nakreslit na číselnou osu)
- ačkoliv je výsledná délka rovna 0, počet bodů, který Cantorovo diskontinuum obsahuje, je nekonečně mnoho
- je jich přesně tolik, kolik bylo čísel v původním intervalu $(0;1)$

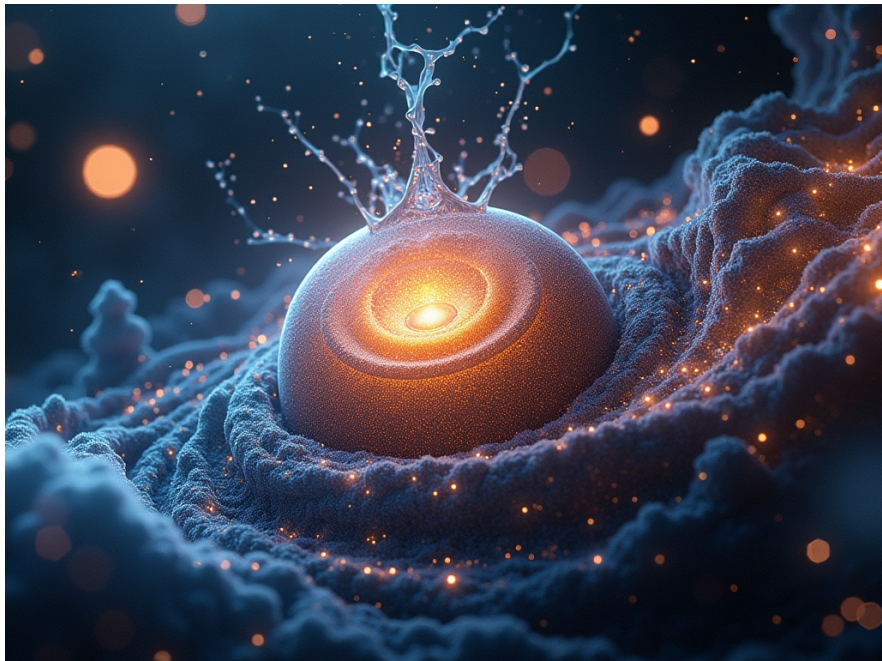


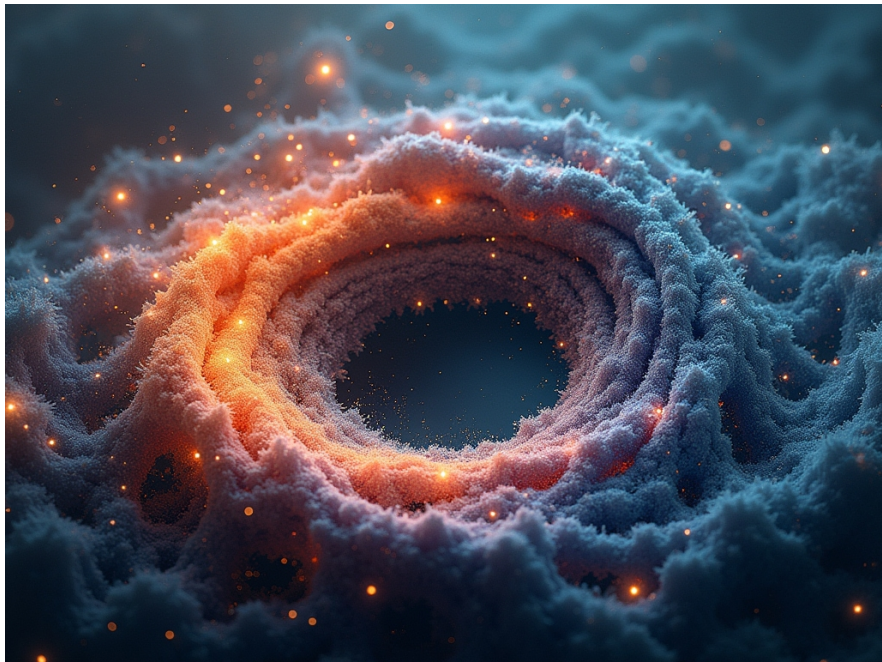












Děkuji za pozornost.