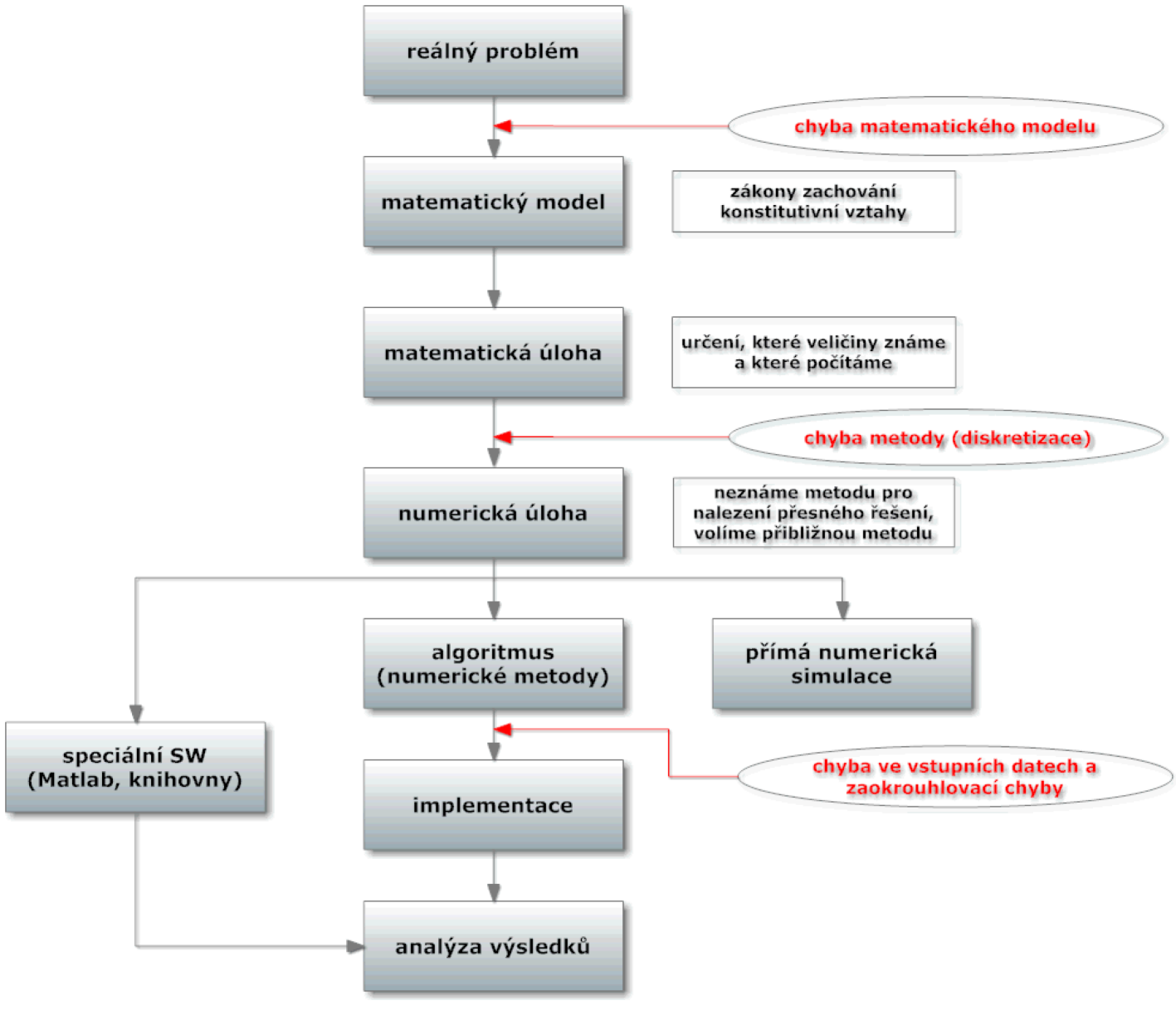
Kapitola 1. Úvod do numerické matematiky

Numerická matematika = věda, která se zabývá řešením matematicky formulovaných úloh pomocí logických operací a aritmetických operací s čísly o konečné délce.



Příklad

Reálný problém ... intravenózní dávkování léku

Matematický model

- nezávisle proměnná je pouze čas t
- šíření látky není závislé na prostorových proměnných

- popis pomocí diferenciální rovnice

$$\boxed{\frac{dC}{dt} = -k \cdot C}$$

kde C je koncentrace látky v krvi a $k > 0$ je absorpční koeficient

- počáteční podmínka

$$\boxed{C(0) = C_0}$$

chyba matematického modelu odpovídá zjednodušujícím předpokladům

Matematická úloha

- chceme vypočítat hodnotu koncentrace látky v čase $t \in < 0, T >$

Numerická úloha

- řešení hledáme pouze v konečně mnoha bodech
  (diskretizujeme čas, $t_0 = 0$, $t_n = n \cdot \frac{T}{N}$, $t_N = T$)
  N je počet dělení intervalu $< 0, T >$

chyba diskretizace (metody)

Numerická metoda

- derivaci $\frac{dC}{dt}$ aproximujeme poměrnou diferencí

$$\boxed{\frac{C_{n+1} - C_n}{\frac{T}{N}} = -k \cdot C_n}$$

chyba diskretizace (metody)

Výpočet

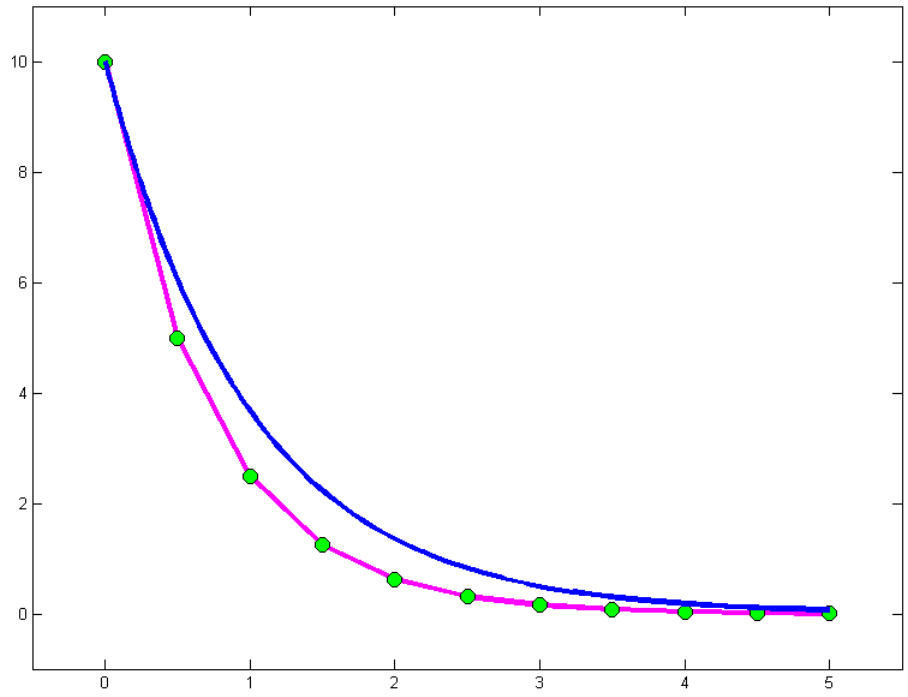
$$C_{n+1} = (1 - \frac{T}{N} \cdot k) \cdot C_n, \quad C_0 \text{ dáno}$$

zaokrouhlovací chyby

Analytické řešení

$$C(t) = C_0 \cdot e^{-kT}$$

např: $C(0) = 10, \quad k = 1, \quad T = 5, \quad N = 10$



Reálné hodnoty ???

CHYBY

x ... přesná hodnota

$\tilde{x}$ ... přibližná hodnota

absolutní chyba ... $A(x) = |x - \tilde{x}| \leq \underbrace{a(x)}_{odhad}$

relativní chyba ... $R(x) = \dfrac{A(x)}{|x|} \leq \underbrace{r(x)}_{odhad}$

Pozn.: Při odečítání „blízkých“ čísel roste relativní chyba (ztráta platných číslic)

$$\boxed{a(x \pm y) = a(x) + a(y)}$$

$$|(x \pm y) - (\tilde{x} \pm \tilde{y})| \leq |x - \tilde{x}| + |\tilde{y} - y|$$

$$\boxed{r(x \pm y) = \frac{a(x) + a(y)}{|x \pm y|}} \qquad |x \pm y| \to 0_+ \quad !!!$$

Pozn.: Násobení a dělení nemohou podstatně zvětšit relativní chybu

$$\boxed{a(x \cdot y) = |x| \cdot a(y) + |y| \cdot a(x)}$$

$$|xy - \tilde{x}\tilde{y}| = |xy - \tilde{x}y + \tilde{x}y - \tilde{x}\tilde{y}| = |y(x - \tilde{x}) + \underbrace{\tilde{x}}_{\approx x}(y - \tilde{y})| \leq |y| \cdot |x - \tilde{x}| + |x| \cdot |y - \tilde{y}|$$

$$\boxed{r(x \cdot y) = r(x) + r(y)}$$

$$\frac{|x|a(y)+|y|a(x)}{|xy|}=\frac{a(y)}{|y|}+\frac{a(x)}{|x|}$$

$$\boxed{a\left(\frac{x}{y}\right)=\frac{|x|\cdot a(y)+|y|\cdot a(x)}{y^2}}$$

$$\left|\frac{x}{y}-\frac{\tilde{x}}{\tilde{y}}\right|=\left|\frac{1}{y\tilde{y}}(x\tilde{y}-\tilde{x}y)\right|=\left|\frac{1}{y\underbrace{\tilde{y}}_{\approx y}}(x\tilde{y}-xy+xy-\tilde{x}y)\right|=$$

$$=\left|\frac{1}{y\tilde{y}}\left(x(\tilde{y}-y)+y(x-\tilde{x})\right)\right|\leq\frac{1}{y^2}(|x|\cdot|y-\tilde{y}|+|y|\cdot|x-\tilde{x}|)$$

$$\boxed{r\left(\frac{x}{y}\right)=r(x)+r(y)}$$

$$\frac{\frac{|x|a(y)+|y|a(x)}{y^2}}{\left|\frac{x}{y}\right|}=\frac{a(y)}{|y|}+\frac{a(x)}{|x|}$$

Definice: Mějme dány dvě množiny X(vstupní data) a Y(výstupní data). Předpokládejme, že X, Y jsou Banachovy prostory. **Úlohou** rozumíme relaci

$$y=U(x),\ x\in X,\ y\in Y.$$

Definice: Řekneme, že úloha je **korektní** na dvojici prostorů (X,Y), když

- $\forall x\in X\ \exists!\, y\in Y:\ y=U(x)$ (zobrazení),
- řešení y spojitě závisí na vstupních datech

$$\forall\{x_n\}:\ x_n\to x,\ U(x_n)=y_n:\ y_n\to y=U(x).$$

Poznámka: Banachův prostor = úplný + normovaný

úplný prostor: metrický prostor, kde $\forall$ Cauchyovská posl. $u_n\subset X$ má limitu $u\in X$

normovaný prostor = množina X:

a) X je lineární;

b) $\forall u\in X\to\|u\|$:

$$\|u\|\geq 0,\ \ \|u\|=0\Leftrightarrow u=0;$$
$$\|au\|=|a|\cdot\|u\|\quad\forall a\in\mathbb{R};$$
$$\|u+v\|\leq\|u\|+\|v\|;$$

c) $d(u,v)=\|u-v\|$

Poznámka: Protože X,Y jsou Banachovy prostory, lze spojitost zaručit podmínkou

$$\|y_n-y\|_Y\leq L\|x_n-x\|_X.$$

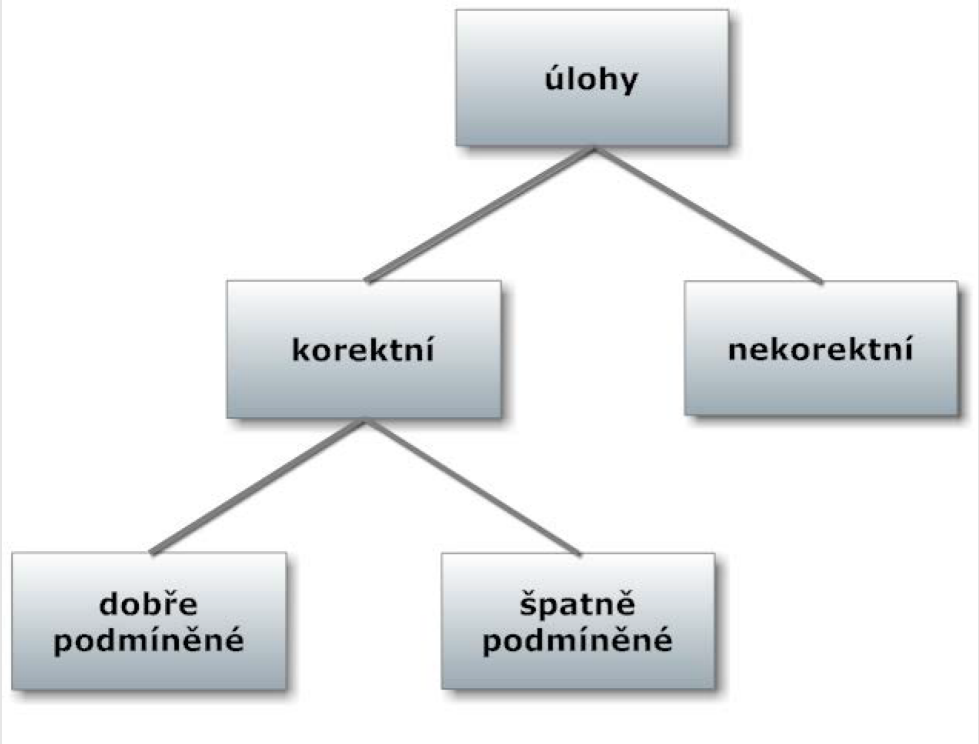
Poznámka: **Nekorektní** úlohy jsou úlohy, které nejsou korektní.
Někdy je nekorektnost způsobena pouze nevhodnou formulací.

Definice: Úloha je **dobře podmíněná**, jestliže malá relativní změna ve vstupních datech vyvolá malou relativní změnu řešení.

Číslo podmíněnosti úlohy $y = U(x)$

$$C_p = \frac{\frac{\|\Delta y\|}{\|y\|}}{\frac{\|\Delta x\|}{\|x\|}}$$

Poznámka: Je-li $C_p \approx 1$ je úloha velmi dobře podmíněná.
V praxi hovoříme o špatně podmíněné úloze pro $C_p \gtrapprox 100$.



Příklad 1

Posuďte podmíněnost úlohy určit hodnotu funkce $y = \sin(x)$

a) v bodě 3,14;
b) v bodě -0,01.

---

a) Volíme $x = 3,14, \quad \Delta x = 0,01 \quad \leftarrow \quad$ *malá změna na vstupu*

$(y =) \ \sin x = \sin 3,14 = 0,0015926$

$(y + \Delta y =) \ \sin(x + \Delta x) = \sin 3,15 = -0,0084072$

$\Delta y = \sin(x + \Delta x) - \sin x = -0,0099998 \quad \leftarrow \quad$ *změna na výstupu*

Relativní chyba na vstupu: $\frac{|\Delta x|}{|x|} \doteq 0,0031847$

Relativní chyba na výstupu: $\dfrac{|\Delta y|}{|y|} \doteq 6,2789149$

$C_p \doteq 1971,6 \quad \rightarrow \quad$ špatně podmíněná úloha

---

b) Volíme $x = -0,01,\ \Delta x = 0,01$

$\sin x = -0,0099998$

$\sin(x + \Delta x) = \sin 0 = 0$

$\Delta y = 0,099998$

Relativní chyba na vstupu: $\dfrac{|\Delta x|}{|x|} \doteq 1$

Relativní chyba na výstupu: $\dfrac{|\Delta y|}{|y|} \doteq 1$

$C_p \doteq 1 \quad \rightarrow \quad$ velmi dobře podmíněná úloha

Poznámka: Podívejme se na předchozí příklad obecněji. Úloha má tvar $y = f(x)$.

Podle věty o střední hodnotě platí:

$$|\Delta y| \approx |f'(x)| \cdot |\Delta x|$$

odtud:

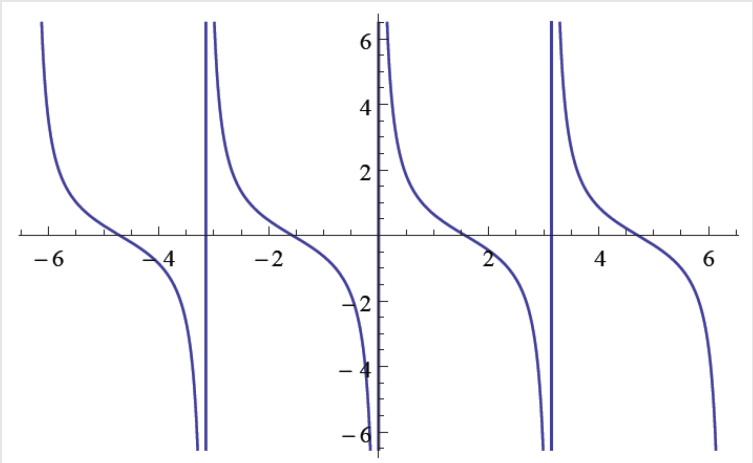
$$\left|\frac{\Delta y}{y}\right| \approx \frac{|f'(x)| \cdot |\Delta x|}{|f(x)|} = \boxed{\left|\frac{x \cdot f'(x)}{f(x)}\right|} \cdot \left|\frac{\Delta x}{x}\right|$$

Tedy

$$C_p \approx \left|\frac{x \cdot f'(x)}{f(x)}\right|$$

v našem případě: $y = \sin x \quad \Rightarrow \quad y' = \cos x$

$$C_p \approx \left|\frac{x \cos x}{\sin x}\right| = |x \operatorname{cotg} x|$$



$$\lim_{x \to \pi\pm} x \operatorname{cotg} x = \pm\infty$$

$$\lim_{x \to 0\pm} x\frac{\cos x}{\sin x} = \cos 0 \cdot \lim_{x \to 0} \frac{x}{\sin x} = 1 \cdot 1 = 1$$

Poznámka: Podobné příklady (posuďte podmíněnost úlohy určit hodnotu):

a) $\boxed{f(x) = x^\alpha,\ x \to 0,\ x>0}$

$$C_p = \left|\frac{x f'(x)}{f(x)}\right| = \left|\frac{x\alpha x^{\alpha-1}}{x^\alpha}\right| = \alpha$$

b) $\boxed{f(x) = \arcsin x,\ x \to 1,\ x < 1}$

$$C_p = \left|\frac{x f'(x)}{f(x)}\right| = \left|\frac{x\frac{1}{\sqrt{1-x^2}}}{\arcsin x}\right| = \left|\underbrace{\frac{1}{\sqrt{1-x^2}}}_{\to 0+}\overbrace{\left(\frac{x}{\arcsin x}\right)}^{\to 1}\right| \xrightarrow{x\to 1} \infty$$

c) $\boxed{f(x) = x - 1,\ x \to 1}$

$$C_p = \left|\frac{x \cdot 1}{x - 1}\right| \longrightarrow \infty$$

Příklad 2

Posuďte podmíněnost úlohy řešit soustavu lineárních algebraických rovnic (pro $\alpha \neq \pm 1$)

$$\begin{aligned} x + \alpha y &= 1 \\ \alpha x + y &= 0 \\ x(1-\alpha^2) &= 1 \end{aligned}$$

$$\boxed{x = \frac{1}{1-\alpha^2}}$$

$$y = -\frac{\alpha}{1-\alpha^2}$$

Nechť vstup je hodnota $\underline{\alpha}$ a výstup hodnota $\underline{x}$.
Pak

$$C_p = \frac{\frac{|\Delta x|}{|x|}}{\frac{|\Delta\alpha|}{|\alpha|}} \overset{*}{\approx} \boxed{\left|\alpha\frac{\frac{dx}{d\alpha}}{x}\right|} \underbrace{=}_{**} \left|\alpha\frac{\frac{2\alpha}{(1-\alpha^2)^2}}{\frac{1}{1-\alpha^2}}\right| = \frac{2\alpha^2}{1-\alpha^2}$$

$\Rightarrow$ pro $\alpha^2 \to 1$ je tato úloha špatně podmíněná!

---

* viz předchozí poznámka

** $\frac{dx}{d\alpha} = \frac{d}{d\alpha}\left(\frac{1}{1-\alpha^2}\right) = -\left(\frac{1}{(1-\alpha^2)^2}(-2\alpha)\right)$

Pozn.: Matice výše uvedené soustavy je pro hodnoty α blízké ± 1 skoro singulární.

STABILITA (PODMÍNĚNOST) ALGORITMU

„U nestabilní metody (algoritmu) se relativně malé chyby v jednotlivých krocích výpočtu postupně akumulují tak, že dojde ke katastrofální ztrátě přesnosti numerického řešení úlohy."

- Při výpočtu dochází k zaokrouhlovacím chybám. Je proto vhodné vybírat algoritmy málo citlivé na zaokrouhlovací chyby.

Stabilní algoritmus

- dobře podmíněný - málo citlivý na poruchy ve vstupních datech
- numericky stabilní - málo citlivý na vliv zaokrouhlovacích chyb

Poznámka:

U stabilních metod roste chyba výsledku s počtem kroků N nejvýše lineárně
(v ideálním případě, kdy je znaménko chyby náhodné, zaokrouhlovací chyba roste $\sim \sqrt{N}$).

U nestabilních metod roste zaokrouhlovací chyba rychleji, např. geometrickou řadou $\sim q^N$, kde $|q| > 1$.

Příklad 3

Řešte diferenční rovnici (rekurentní formule, nestabilní rekurze)

$$x_{n+1} = \frac{13}{3}x_n - \frac{4}{3}x_{n-1}, \quad x_0 = 1,\ x_1 = \frac{1}{3}$$

Snadno se ukáže, že řešení je $x_n = \frac{1}{3^n}$ (dosazením).

Při numerickém výpočtu dojdeme k problémům (viz obr). Hodnoty x_n začnou velmi rychle klesat.
Pro vysvětlení ukážeme obecné řešení zadané diferenční rovnice.

- charakteristický polynom

$$\lambda^2 = \frac{13}{3}\lambda - \frac{4}{3}$$

(předpokládáme řešení λ^n: $\lambda^{n+1} = \frac{13}{3}\lambda^n - \frac{4}{3}\lambda^{n-1}$)

- kořeny

$$\lambda_{1,2} = \frac{\frac{13}{3} \pm \sqrt{(\frac{13}{3})^2 - 4 \cdot \frac{4}{3}}}{2} = \frac{\frac{13}{3} \pm \sqrt{\frac{121}{9}}}{2}, \quad \text{tj. } \lambda_1 = \frac{1}{3}, \lambda_2 = 4$$

- obecné řešení

$$x_n = A \cdot \left(\frac{1}{3}\right)^n + B \cdot 4^n$$

$$x_0 = 1 = A \cdot \left(\frac{1}{3}\right)^0 + B \cdot 4^0 = A + B = 1$$
$$x_1 = \frac{1}{3} = A \cdot \left(\frac{1}{3}\right)^1 + B \cdot 4^1 = \frac{1}{3} \cdot A + 4 \cdot B = \frac{1}{3}$$
$$\Rightarrow \quad A = 1,\ B = 0$$

Přes počáteční podmínku $B = 0$ vzniknou vlivem zaokrouhlovacích chyb malé druhé komponenty řešení

Výsledky z MATLABu, FORMAT SHORT, pevná čárka na 5 číslic

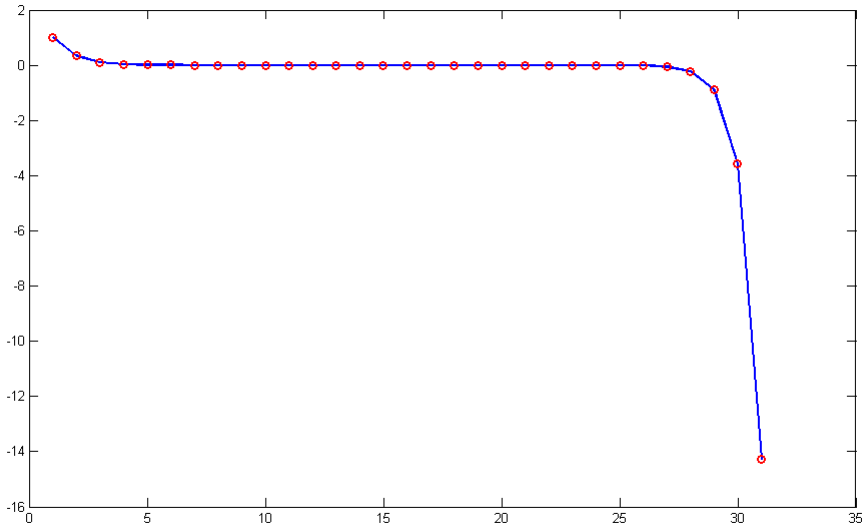
```
clc;
clear;
format short;

n=30;

x(1)=1;
x(2)=1/3;

for i=2:n
 x(i+1)=13/3*x(i)-4/3*x(i-1);
end

plot(1:n+1,x,'b-',1:n+1,x,'ro');
```



Příklad 4

Vypočtěte přibližně hodnotu

$$J_n = \int_0^1 \frac{x^n}{x+5}\,dx$$

Platí:

$$\underbrace{\int_0^1 x^{n-1}\,dx}_{\left[\frac{1}{n}x^n\right]_0^1 = \frac{1}{n}} = \int_0^1 \frac{x^{n-1}(x+5)}{x+5}\,dx = \underbrace{\int_0^1 \frac{x^n}{x+5}\,dx}_{J_n} + 5\underbrace{\int_0^1 \frac{x^{n-1}}{x+5}\,dx}_{J_{n-1}}$$

Dále:

$$J_0 = \int_0^1 \frac{1}{x+5}\,dx = \left[\ln|x+5|\right]_0^1 = \ln\frac{6}{5}$$

Rekurentní formule:

$$\boxed{\begin{aligned} J_0 &= \ln\frac{6}{5} \\ J_n &= -5 \cdot J_{n-1} + \frac{1}{n} \end{aligned}}$$

Nestabilní algoritmus ! ... vždy $\exists n_0 : \; J_{n_0} < 0$!

Proto je lépe postupovat odzadu:

- dokážeme, že $\boxed{\lim_{n\to\infty} J_n = 0}$

$$|J_n| = \left| \int_0^1 \frac{x^n}{x+5}\,dx \right| \le \int_0^1 \left| \frac{x^n}{x+5} \right| dx \le \frac{1}{5}\int_0^1 x^n dx = \frac{1}{5(n+1)} \to_{n\to\infty} 0$$

- např. zvolíme $J_{100} = 0$ a počítáme $J_{n-1} = -\frac{1}{5}(J_n - \frac{1}{n})$

$$\boxed{\begin{aligned} J_{100} &= 0 \\ J_{n-1} &= -\frac{1}{5} \cdot J_n + \frac{1}{5n} \end{aligned}}$$

Výsledky z MATLABu, FORMAT SHORT E

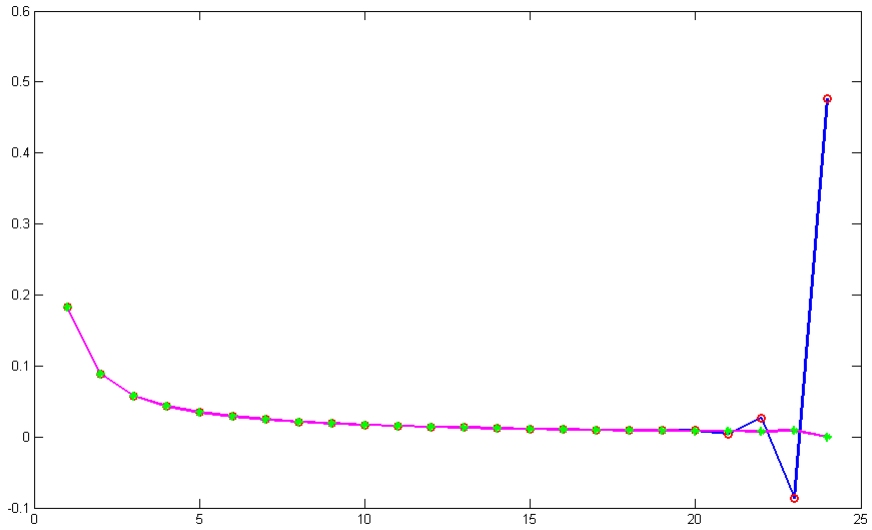
```
clc;
clear;
format short e;

n=24;

J(1)=log(6/5);
JJ(n)=0;

for i=1:n-1
 J(i+1) = -5*J(i) + 1/i;
end
for i=n-1:-1:1
 JJ(i) = (1/i - JJ(i+1)) / 5;
end

[J' JJ']
plot(1:n,J,'b-',1:n,J,'ro');
hold on
plot(1:n,JJ,'m-',1:n,JJ,'g*');
```



J	JJ
1.8232e-001	1.8232e-001
8.8392e-002	8.8392e-002
5.8039e-002	5.8039e-002
4.3139e-002	4.3139e-002
3.4306e-002	3.4306e-002
2.8468e-002	2.8468e-002
2.4325e-002	2.4325e-002
2.1233e-002	2.1233e-002
1.8837e-002	1.8837e-002
1.6926e-002	1.6926e-002
1.5368e-002	1.5368e-002
1.4071e-002	1.4071e-002
1.2977e-002	1.2977e-002
1.2040e-002	1.2040e-002
1.1229e-002	1.1229e-002
1.0522e-002	1.0521e-002
9.8903e-003	9.8964e-003
9.3719e-003	9.3414e-003
8.6960e-003	8.8485e-003
9.1515e-003	8.3893e-003
4.2426e-003	8.0535e-003
2.6406e-002	7.3518e-003
-8.6575e-002	8.6957e-003
4.7635e-001	0

Zobrazení čísel

Motivace: $\sum_{k=1}^{100000} \frac{1}{10} = 9998{,}55664$

- Lidé používají desítkovou soustavu.
- Počítače dvojkovou.

Komunikace s počítačem

- Zadání v 10-soustavě.
- Převod do 2-soustavy (počítač).
- Výpočet (počítač).
- Zpětný převod do 10-soustavy (počítač).
- Výsledek v 10-soustavě.

Soustavy

- desítková

  $1563 = (1 \cdot 10^3) + (5 \cdot 10^2) + (6 \cdot 10^1) + (3 \cdot 10^0)$

  obecně

  $N = (a_k \cdot 10^k) + (a_{k-1} \cdot 10^{k-1}) + \cdots + (a_1 \cdot 10^1) + (a_0 \cdot 10^0)$

  $(N \in \mathbb{N}), \quad a_k \in \{0, 1, 2, \ldots, 9\}$

  značení

  $N = a_k a_{k-1} a_{k-2} \ldots a_1 a_0$

- dvojková

  $1563 = (1 \cdot 2^{10}) + (1 \cdot 2^9) + (0 \cdot 2^8) + (0 \cdot 2^7) + (0 \cdot 2^6) + (0 \cdot 2^5) + (1 \cdot 2^4) + (1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (1 \cdot 2^0)$

$$\boxed{(1563)_{10} \;=\; (11000011011)_2}$$

Binární zlomky

lze vyjádřit jako sumu se zápornými mocninami dvou

$R \in \mathbb{R} \qquad 0 < R < 1 \qquad d_j \in \{0, 1\}$

$R = (d_1 \cdot 2^{-1}) + (d_2 \cdot 2^{-2}) + \ldots + (d_n \cdot 2^{-n}) + \ldots$

$R = (0, d_1\, d_2\, \ldots\, d_n\, \ldots)_2$

Zápis čísel

- V desítkové soustavě (vědecká notace)

  $0,000747 = 7,47 \cdot 10^{-4}$

  $313,815 = 3,13815 \cdot 10^2$

- Strojová čísla

  normalizovaná pohyblivá řádová čárka (REAL)

$$\boxed{\mathbf{x} = \pm\, \mathbf{q} \cdot \mathbf{2^n} \qquad \tfrac{1}{2} \leq \mathbf{q} < 1 \ldots \text{mantisa}, \quad \mathbf{n} \ldots \text{exponent}}$$

Poznámka: Mnoho reálných čísel, které lze v desítkové soustavě zapsat pomocí konečného počtu cifer, pro zápis ve dvojkové soustavě vyžaduje nekonečně mnoho cifer.

$$(0,7)_{10} = (0,1\overline{0110})_2 = 1\cdot 2^{-1} + \sum_{k=0}^{\infty} 1\cdot 2^{-(3+4k)} + \sum_{k=0}^{\infty} 1\cdot 2^{-(4+4k)} =$$

$$= 2^{-1} + 2^{-3}\cdot\sum_{k=0}^{\infty}\left(2^{-4}\right)^k + 2^{-4}\cdot\sum_{k=0}^{\infty}\left(2^{-4}\right)^k = \frac{1}{2} + \frac{1}{8}\cdot\underbrace{\frac{1}{1-\frac{1}{16}}}_{=\frac{16}{15}} + \frac{1}{16}\cdot\frac{16}{15} =$$

$$= \frac{1}{2} + \frac{2}{15} + \frac{1}{15} = \frac{15+4+2}{30} = \frac{21}{30} = \frac{7}{10}$$

Příklad:

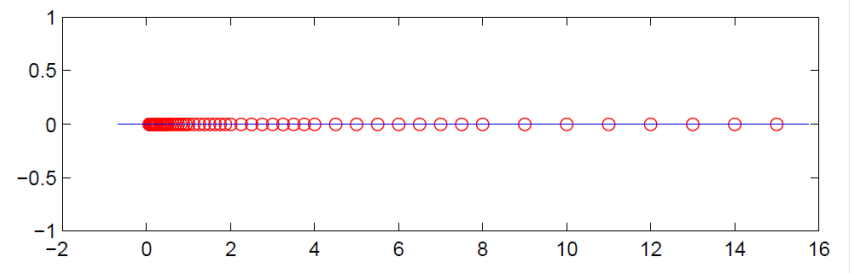
Sestrojte všechna strojová čísla s mantisou délky 4 a exponentem v rozsahu od -3 do 4, tj.

$x = \mathbf{q}\cdot 2^{\mathbf{n}}$, kde $\mathbf{q} = 0,d_1\,d_2\,d_3\,d_4$, $\mathbf{n}\in\{-3,-2,-1,0,1,2,3,4\}$

Abychom si lépe uvědomili jakou mantisou a jakým exponentem je určeno získané číslo, uvedeme si je v následující tabulce.

$\mathbf{q}\setminus\mathbf{n}$	-3	-2	-1	0	1	2	3	4
0.1000_2	0,0625	0,125	0,25	0,5	1	2	4	8
0.1001_2	0.0703125	0,140625	0,28125	0,5625	1,125	2 ,25	4,5	9
0.1010_2	0.078125	0,15625	0,3125	0,625	1,25	2 ,5	5	10
0.1011_2	0.0859375	0,171875	0,34375	0,6875	1,375	2 ,75	5,5	11
0.1100_2	0.09375	0,1875	0,375	0,75	1,5	3	6	12
0.1101_2	0.1015625	0,203125	0,40625	0,8125	1,625	3, 25	6,5	13
0.1110_2	0.109375	0,21875	0,4375	0,875	1,75	3, 5	7	14
0.1111_2	0.1171875	0,234375	0,46875	0,9375	1,875	3, 75	7,5	15

Získaná čísla si je také vhodné vykreslit na číselnou osu, získáme tak přehled o jejich rozložení. Snadno zjistíme, že čísla nejsou rozložena rovnoměrně.



pomocná funkce v MATLABu

```
function [A,P]=stroj_cisla(cisel_mantisy,exponent,zobraz);
%
% [A,P]=stroj_cisla(4,-3:4,1);

for i=1:length(exponent)
 for j=0:2^(cisel_mantisy-1)-1
   zaklad=dec2bin(j);
   zakladstr=num2str(zaklad);
   for k=1:cisel_mantisy-length(zakladstr)-1
     zakladstr=strcat('0',zakladstr);
   end;
   zakladstr=strcat('1',zakladstr);
   zaklad=bin2dec(zakladstr)*2^(-cisel_mantisy);
   A(j+1,i)=zaklad*2^exponent(i);
 end;
end;

[k,l]=size(A);
P=sort(reshape(A,1,k*l));

if zobraz==1
  figure(1);
  plot(P,zeros(size(P)),'ro');
  pr=(P(k*l)-P(1))/20;
  hold on;
  plot([P(1)-pr,P(k*l)+pr],[0 0],'b-');
end;

format short g;
```

Příklad 5:

Uvažujme množinu strojových čísel vygenerovanou v předchozím příkladu (tj. strojová čísla s mantisou délky 4 a exponentem v rozsahu od -3 do 4). Předpokládáme, že počítač zobrazí číslo na nejbližší číslo, které lze zobrazit, v případě shody na větší.

Ukažme si, jak se v tomto stroji sečtou čísla $\frac{1}{10}$ a $\frac{1}{5}$.

Výsledky získané z MATLABu

```
-----------------------------------------------------------------
 Zobrazeni souctu cisel A a B v zadane mnozine strojovych cisel
 s mantisou delky M a exponentem v rozsahu od Exp_min do Exp_max
-----------------------------------------------------------------
 Cislo A = 0.100000
 Cislo B = 0.200000
 Pocet cisel mantisy M = 4
 Rozsah pro exponent:  od -3 do 4

  cislo              zapis obrazu       obraz
-----------------------------------------------------------------
A =  0.100000        0.1101 x 2^-3      0.1015625
B =  0.200000        0.1101 x 2^-2      0.203125

obrazA+obrazB=
     0.3046875       0.1010 x 2^-1      0.3125

A+B= 0.300000        0.1010 x 2^-1      0.3125
```

Poznámka:

V tomto příkladě se shodoval obraz přesného výsledku s obrazem součtu obrazů jednotlivých sčítanců.

Příklad 6:

Uvažujme množinu strojových čísel vygenerovanou v předchozím příkladu (tj. strojová čísla s mantisou délky 4 a exponentem v rozsahu od -3 do 4). Předpokládáme, že počítač zobrazí číslo na nejbližší číslo, které lze zobrazit, v případě shody na větší.

Ukažme si, jak se v tomto stroji sečtou čísla $\frac{3}{10}$ a $\frac{1}{6}$.

Výsledky získané z MATLABu

```
-----------------------------------------------------------------
 Zobrazeni souctu cisel A a B v zadane mnozine strojovych cisel
 s mantisou delky M a exponentem v rozsahu od Exp_min do Exp_max
-----------------------------------------------------------------
 Cislo A = 0.300000
 Cislo B = 0.166667
 Pocet cisel mantisy M = 4
 Rozsah pro exponent:  od -3 do 4

  cislo              zapis obrazu       obraz
-----------------------------------------------------------------
A =  0.300000        0.1010 x 2^-1      0.3125
B =  0.166667        0.1011 x 2^-2      0.171875

obrazA+obrazB=
     0.484375        0.1000 x 2^0       0.5

A+B= 0.466667        0.1011 x 2^0       0.46875
```

Poznámka:

V tomto příkladě se obraz přesného výsledku s obrazem součtu obrazů jednotlivých sčítanců neshodoval !

Chyba výpočtu:

$$\frac{7}{15} - 0,1000_2 \cdot 2^0 = \frac{14-15}{30} = -\frac{1}{30} = -0,0\overline{3}$$

Relativně:

$$\frac{\frac{1}{30}}{\frac{7}{15}} = \frac{1}{14} = 7,14\,\% \qquad !!!$$

Přesnost počítače

- Vymezíme-li pro mantisu 24 bitů, získáme 7 desetinných míst ($2^{24} = 16\,777\,216$).
- Vymezíme-li pro mantisu 32 bitů, získáme 9 desetinných míst ($2^{32} = 4\,294\,967\,296$).

Základní formáty:

Formát	Bytes	Bitů pro mantisu	Bitů pro exponent
Single	4	24	8
Double	8	53	11

Příklad:

- Uvažujme formát SINGLE , tj. 24 bitů pro mantisu.

  $\frac{1}{10} = 0,0\overline{0011}_2 \approx 0,1100\,1100\,1100\,1100\,1100\,1100_2 \cdot 2^{-3}$.

  Chyba zobrazení je $0,\overline{1100}_2 \cdot 2^{-27} (= \frac{1}{10} \cdot 2^{-24}) \approx 5,96 \cdot 10^{-9}$.

- Máme-li počítat $\sum_{k=1}^{100000} \frac{1}{10}$, dostaneme ve formátu SINGLE 9.998, 55664.

  Chyba musí být větší než $100000 \cdot 5,96 \cdot 10^{-9} = \mathbf{5,96 \cdot 10^{-4}}$.

Ve skutečnosti je chyba ještě větší, neboť se v průběhu výpočtu musí částečně suma zaokrouhlovat dolů nebo nahoru, jak suma roste, později přičítaná čísla $\frac{1}{10}$ jsou oproti sumě menší a jsou tedy počítány s menší přesností (viz následující příklad).

Příklad: Ve formátu SINGLE sečtěte čísla 10000 a 0,1.

```
------------------------------------------------------------------------
Prevod cisla 10000 z 10-soustavy do 2-soustavy na 0 desetinnych mist
Cela cast ........... 10000
Desetinna cast ...... 0.000000
------------------------------------------------------------------------
prevod_cele_casti =

10000 : 2 = 5000 : 2 = 2500 : 2 = 1250 : 2 = 625 : 2 = 312 : 2 =
    0          0          0          0         1         0


= 156 : 2 = 78 : 2 = 39 : 2 = 19 : 2 = 9 : 2 = 4 : 2 = 2 : 2 = 1
    0       0       1       1       1       0       0

Cislo 10000 v 10-soustave prevedeno do 2-soustavy je 10011100010000.
```

$$(10000)_{10} = (10011100010000)_2 = 0,100111001 \cdot 2^{14}$$

$$(2^{14} = 16384)$$

$$\begin{aligned}
10000 \;\ldots\; & 0,1001\,1100\,1000\,0000\,0000\,0000 \cdot 2^{14} \\
0,1 \;\ldots\; & 0,1100\,1100\,1100\,1100\,1100\,1100 \cdot 2^{-3} \\
0,1 \text{ po SHIFTu} \;\ldots\; & 0,0000\,0000\,0000\,0000\,0110\,0110 \cdot 2^{14} \\
& = (01100110)_2 \cdot 2^{-24} \cdot 2^{14} = (64+32+4+2) \cdot 2^{-10} = \\
& = \tfrac{102}{1024} = 0,099609375
\end{aligned}$$

$$10000 + 0,1 \;\ldots\; 0,1001\,1100\,1000\,0000\,0110\,0110 \cdot 2^{14}$$

Číslo 10000 je zobrazeno přesně.
Chyba zobrazení 0,1 po SHIFTU je $\frac{1}{10} - \frac{102}{1024} = 0,1 - 0,099609375 = 3,90625 \cdot 10^{-4}$

Shrnutí:

$$10000 + 0,1 \;\rightarrow\; \text{výsledek s chybou } 3,90625 \cdot 10^{-4}$$

(v sumě z motivačního příkladu jde o jeden krok)

skript v MATLABu

```
s=0;
h=single(1/10);

for i=1:100000
  s=s+h;
end;

s
```