

# Rigorózní výpočty s nepřesnými čísly a matematické důkazy

Petr Girg, Plzeň

Registrační číslo projektu: **CZ. 1.07/2.2.00/07.0332**

Název projektu: **Matematika pro inženýry 21. století - inovace výuky matematiky na technických školách v nových podmínkách rychle se vyvíjející informační a technické společnosti**



evropský  
sociální  
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání  
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

## Číslo v počítači

Takzvaná strojem podporovaná čísla jsou čísla, která mají stejný počet binárních číslic jako je velikost registru procesoru a operace s nimi jsou podporovány přímo instrukční sadou daného procesoru. Výpočty se strojem podporovanými čísly jsou tedy nejefektivnější z hlediska výpočetního času i paměťové náročnosti.

Cílem této přednášky je stručně vysvětlit zdroje chyb při výpočtech v počítačové aritmetice a ukázat, jak počítačovou aritmetiku použít v rigorózních důkazech.

### Třídy uživatelů

*Absolutně optimistický uživatel:* „každý výsledek spočtený počítačem je absolutně správný.“

*Informovaný uživatel:* „některé výsledky vypočtené na počítači mohou být chybné, tyto případy jsou ale velmi řídké a spíše uměle vykonstruované.“

*Absolutně pesimistický uživatel:* „úplně každý výsledek vypočtený na počítači je pouze aproximací (tedy je matematicky nesprávný), neboť počítače používají přibližná čísla.“

*Pokročilý uživatel:* „některé výsledky jsou matematicky nesprávné některé jsou správné. Pokročilý uživatel si je vědom faktu, že počítačová aritmetika se řídí jistými pravidly a správnost výsledku je předem určena těmito pravidly. Chyba ve výpočtu nevzniká náhodně nějakou magickou mocí.“

*Expertní uživatel:* zná pravidla jimiž se počítačová aritmetika řídí a dokáže se ke stejnému výsledku jako počítač dopracovat ručním výpočtem.

### Další třída (ne již uživatelská)

*Chip developer:* ví, jak je čip zadrátkován a tudíž ví také o vyjimečných situacích, kdy se čip nechová tak, jak je popsáno v manuálu (např. vliv teploty, tlaku, magnetického pole, erupcí na slunci a kosmického záření atd. na funkčnost čipu).

### Hlavní zdroje chyb:

Počítač má konečně mnoho stavů. Tudíž může uchovávat a počítat s konečně mnoha čísly, reprezentovatelnými konečně mnoha binárními číslicemi.

a) Typ **Integer** na 32-bitovém počítači: celá čísla v rozsahu od  $-2^{31}$  do  $2^{31} - 1$ .

Aritmetické operace sčítání, odčítání pro tato čísla jsou přesná, pokud nedojde k přetečení (viz níže).

Omezení typu Integer - malý rozsah.

Např.

$$12! < 2^{31} - 1 < 13!,$$

$$10^9 < 2^{31} - 1 < 10^{10}.$$

b) Typ **Real** v moderních počítačích (podle normy IEEE 754) je konečný výběr z množiny reálných čísel. Všechna čísla tohoto typu jsou dokonce racionální čísla. V jazyce C se označují jako typ **float**, **double**, **extended double** a podobně.

Tyto konečné množiny se používají v numerické matematice. Numerická matematika se z velké části zabývá řešením počátečních a okrajových úloh pro diferenciální rovnice. Tedy konečné diskrétní množiny se používají k modelování spojitých matematických objektů, jejichž teoretické vlastnosti jsou založeny na Archimedově vlastnosti, souvislosti a úplnosti množiny reálných čísel (a tedy i její nespočetnosti). Základní pojmy matematické analýzy (limita, derivace, integrál) jsou založeny na faktu, že používáme „spojitá“ reálná čísla, podrobná diskuze viz [T.W. Korner: A companion to analysis. American Mathematical Society, Providence, RI, 2004, ISBN: 0-8218-3447-9]





## Simulace reálných čísel - pohyblivá řádová tečka

### ■ Motivace

Avogadrova konstanta  $N = 6.02252 \times 10^{23}$ ,

Planckova konstanta  $h = 6.6260 \times 10^{-34}$  [J · s].

Potřebujeme číselný systém s velkým rozsahem hodnot.

Na vyjádření jednoho čísla řádu  $10^{57}$  typu Integer bychom potřebovali 24 bytů.

### ■ Příklad aritmetiky s pohyblivou řádovou tečkou (modelový, v desítkové soustavě)

Abychom mohli vysvětlit princip aritmetiky s pohyblivou řádovou tečkou i uživatelům, kteří nejsou zběhlí v dvojkové soustavě, budeme uvažovat následující modelový příklad v desítkové soustavě, který věrně modeluje problém zobrazení v počítači (i když tam máme čísla uložená binárně) viz [Knuth, Art of Computer Programming II, Addison-Wesley, 1997].

Nechť  $S$  je množina všech reálných čísel, která lze zapsat ve tvaru

$$\pm 0. a_{-1} a_{-2} a_{-3} \times 10^{\text{exp}}$$

v desítkové soustavě, kde číslice  $a_{-1} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$  a  $a_{-2}, a_{-3} \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ,  $\text{exp} = 10 e_1 + e_2 - 49$ ,  $-49 \leq \text{exp} \leq 98 - 49 = 49$ , hodnota  $e_1 = e_2 = 9$  je rezervována pro speciální účely.

V tomto systému by nešlo zapsat číslo 0.

Proto se kóduje jako  $\pm 0.000 \times 10^{-49}$  a s těmito dvěma čísly se počítá jako s nulou.

Výhoda toho zápisu je snadná rozpoznatelnost, neboť  $e_1, e_2, a_{-1}, a_{-2}, a_{-3} = 0$ .

Jen pokud  $e_1, e_2, a_{-2}, a_{-3} = 0$  je povolena hodnota  $a_{-1} = 0$ . Číslo 0 budeme v našem systému psát jako 0. (nula s tečkou).

Tudíž např. čísla

$$-12.3 = -0.123 \times 10^2 \quad \text{a} \quad 0.123 = 0.123 \times 10^0$$

jsou prvky  $S$ ,

zatímco čísla 12345 i 0.12345 neleží v  $S$ .

**Pozn.:** Proč musí být  $a_{-1}$  nenulové pro  $e_1 \neq 0 \vee e_2 \neq 0$ ? Protože např.  $0.001 \times 10^0 = 0.100 \times 10^{-2}$ . Tím bychom ztratili jednoznačnost zápisu čísel. Principálně (beze ztráty jednoznačnosti) lze do systému přidat ještě čísla  $\pm 0.0 a_{-2} a_{-3} \times 10^{-49}$ , kterým se říká *subnormální*.

Největší výhoda takto definované množiny  $S$  je její velký rozsah - vzdálenost nejmenšího a největšího čísla a také možnost vyjádření čísel blízko nuly při velmi malých paměťových nárocích: 5 dekadických číslic a dvě znaménka.

Nejmenší a největší číslo vyjádřitelné v tomto systému je

$$-0.999 \times 10^{49} \quad \text{a} \quad 0.999 \times 10^{49}$$

Největší záporné číslo a nejmenší kladné číslo vyjádřitelné v tomto systému je

$$-0.100 \times 10^{-49} \quad \text{a} \quad 0.100 \times 10^{-49}$$

Tohoto rozsahu je dosaženo tím, že sousední čísla nejsou od sebe stejně daleko.

$$\text{Např. } 0.999 \times 10^{49} - 0.998 \times 10^{49} = 0.001 \times 10^{49} = 10^{46}$$

$$\text{zatímco } 0.101 \times 10^{-49} - 0.100 \times 10^{-49} = 10^{-52} = 10^{-52}$$

Tedy systém obsahuje velké mezery mezi velkými sousedními čísly a malé mezery mezi malými sousedními čísly.

Velikost mezery mezi čísly  $0.100 \times 10^{\text{exp}}$  až  $0.999 \times 10^{\text{exp}}$  pro pevně zvolené  $\text{exp} = -49, -48, \dots, 48, 49$  je stejná a je rovna  $0.001 \times 10^{\text{exp}} = 10^{\text{exp}-3}$ .

Následníkem čísla  $0.999 \times 10^{\text{exp}}$  je číslo  $0.100 \times 10^{e+1}$  pro  $e = -49, -8, \dots, 48$ .

Nestejnost těchto mezer má nepříjemný vliv na aritmetické operace prováděné v tomto systému.

$$0.100 \times 10^1 + 0.100 \times 10^1 = 0.200 \times 10^1 \quad \text{Přesný výsledek } 1 + 1 = 2.$$

$0.100 \times 10^1 + 0.100 \times 10^4 = 0.1001 \times 10^4$  Výsledek sčítání neleží v S. Protože uvažujeme pouze čísla z S, musíme výsledek zaokrouhlit k číslu z S:  $0.100 \times 10^4$ . Tím máme matematicky nepřesný výsledek, ale ležící v množině S.

Aby se program v patologických situacích jako je dělení 0, provedení operace jejíž výsledek je větší/menší než největší/nejmenší možná hodnota ležící v S nezasekl a nedošlo k ohrožení stability operačního systému, je účelné do množiny čísel S přidat ještě symboly  $-\infty$ ,  $+\infty$  a NaN, pro vyjádření patologických situací. Výpočet v aritmetice obohacené o tyto symboly pak pokračuje hladce dál vždy a je jen na programátorovi, jak programově ošetří výskyt těchto hodnot ve výsledcích.

$0.999 \times 10^{49} + 0.001 \times 10^{49} = 1.000 \times 10^{50}$  Výsledek není reprezentovatelný v systému S, je příliš velký. Proto jej budeme reprezentovat symbolem  $+\infty$ , symbol  $+\infty$  budeme kódovat se jako  $0.100 \times 10^{50}$ , což odpovídá  $e_1 = e_2 = 9$ .

$0.100 \times 10^0 / 0$ . Výsledek není definován. V tomto případě budeme používat k zápisu výsledku symbol NaN („Not a Number“, to jest výsledek není číslo). Budeme kódovat jako  $0.111 \times 10^{50}$ .

V tomto systému můžeme přibližně vyjádřit

Avogadrovu konstantu  $N \doteq 0.602 \times 10^{24}$  i Planckovu konstantu  $\hbar \doteq 0.662 \times 10^{-33}$  [J · s].

Přičemž všech čísel v tomto systému je jen

$$2 \times 9 \times 10 \times 10 \times 99 \text{ (normální čísla)} + 1 \text{ (nula)} + 2 \text{ (}\pm \text{ nekonečno)} + 1 \text{ (symbol NaN)},$$

to jest celkem 178204 různých čísel a symbolů. ***Cena, kterou za to platíme je nutnost zaokrouhlovat výsledky aritmetických operací.***

**Množina S není uzavřená vůči základním aritmetickým operacím tak jak je běžně chápeme.**

Místo toho jsou definovány operace:

$\oplus$  (součet se zaokrouhlením do S),  $\ominus$  (rozdíl se zaokrouhlením do S),  $\otimes$  (násobení se zaokrouhlením do S) a dělení  $\oslash$  se zaokrouhlením do S.

V tomto číselném systému **neplatí asociativita sčítání**:

$$(0.100 \times 10^1 \oplus 0.100 \times 10^4) \ominus 0.100 \times 10^4 = 0.$$

$$0.100 \times 10^1 \oplus (0.100 \times 10^4 \ominus 0.100 \times 10^4) = 0.100 \times 10^1$$

**Neplatí ani zákon distributivnosti** násobení vůči sčítání:

$$0.200 \times 10^0 \otimes (0.603 \times 10^0 \oplus -0.600 \times 10^0) = 0.600 \times 10^{-3}$$

$$(0.200 \times 10^0 \otimes 0.603 \times 10^0) \oplus (0.200 \times 10^0 \otimes -0.600 \times 10^0) = 0.$$

Výsledek 0 v druhém výpočtu vznikl takto

$$(0.200 \times 10^0 \otimes 0.603 \times 10^0) = \text{round}(0.200 \cdot 0.603) = \text{round}(0.1206) = 0.120 \times 10^0,$$

$$(0.200 \times 10^0 \otimes -0.600 \times 10^0) = -0.120 \times 10^0,$$

kde round označuje zaokrouhlení k nejbližšímu číslu z S nebo jednomu ze symbolů  $\pm\infty$ .

**V množině S neplatí ani řada známých vztahů mezi reálnými elementárními funkcemi.**

Pro nezáporné reálné hodnoty  $x$  platí  $\sqrt{x} \cdot \sqrt{x} = x$ , ale v systému S existují  $x$  nezáporná taková, že  $\text{sqrt}(x) \otimes \text{sqrt}(x) \neq x$  (zde sqrt značí odmocninu zaokrouhlenou k nejbližšímu číslu z S).

Tento vztah neplatí např. pro  $x = 2$ , protože  $\text{sqrt}(0.200 \times 10^1) = 0.141 \times 10^1$  a tedy

$$\text{sqrt}(0.200 \times 10^1) \otimes \text{sqrt}(0.200 \times 10^1) = \text{round}(0.141 \times 10^1 \cdot 0.141 \times 10^1) = \text{round}(1.9881) = 0.199 \times 10^1 \neq 2.$$

Matematicky sice platí  $e^{\ln x} = x$ , ale budeme-li tento výraz vyhodnocovat v S, nemusíme dostat původní hodnotu.

Dosaďme za  $x = 0.200 \times 10^2$ . Dostaneme:

$$\text{round}(\ln(0.200 \times 10^2)) = \text{round}(2.99573) = 0.300 \times 10^1, \quad \text{round}(e^3) = \text{round}(20.0855) = 0.201 \times 10^2 \neq 0.200 \times 10^2.$$

## ■ Binární model

Nechť  $S_2$  je množina všech reálných čísel, která lze zapsat ve tvaru  $\pm 0. a_{-1} a_{-2} a_{-3} \times 10^{\text{exp}}$  v dvojkové soustavě, kde číslice  $a_{-1} = 1$  a  $a_{-2}, a_{-3} \in \{0, 1\}$ ,

$\text{exp} = 2e_1 + e_2 - 1$ ,  $e_1, e_2 \in \{0, 1\}$ ,  $-1 \leq \text{exp} \leq 2 - 1 = 1$ , hodnota  $e_1 = e_2 = 1$  je rezervována pro speciální účely.

Stejně jako v předchozím případě  $a_1 = a_2 = a_3 = e_1 = e_2 = 0$  je použito k reprezentaci 0.

Všechny možnosti vyjádřené dekadicky :

$$\left\{ -\frac{7}{2}, -3, -\frac{5}{2}, -2, -\frac{7}{4}, -\frac{3}{2}, -\frac{5}{4}, -1, -\frac{7}{8}, -\frac{3}{4}, -\frac{5}{8}, -\frac{1}{2}, -\frac{7}{16}, \right. \\ \left. -\frac{3}{8}, -\frac{5}{16}, -\frac{1}{4}, \frac{1}{4}, \frac{5}{16}, \frac{3}{8}, \frac{7}{16}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1, \frac{5}{4}, \frac{3}{2}, \frac{7}{4}, 2, \frac{5}{2}, 3, \frac{7}{2} \right\}$$



Očištěné od hodnot odpovídajících exponentu  $e_1 = e_2 = 1$  :

$$\left\{ -\frac{7}{4}, -\frac{3}{2}, -\frac{5}{4}, -1, -\frac{7}{8}, -\frac{3}{4}, -\frac{5}{8}, -\frac{1}{2}, -\frac{7}{16}, \right. \\ \left. -\frac{3}{8}, -\frac{5}{16}, -\frac{1}{4}, \frac{1}{4}, \frac{5}{16}, \frac{3}{8}, \frac{7}{16}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1, \frac{5}{4}, \frac{3}{2}, \frac{7}{4} \right\}$$

Přidána nula :

$$\left\{ -\frac{7}{4}, -\frac{3}{2}, -\frac{5}{4}, -1, -\frac{7}{8}, -\frac{3}{4}, -\frac{5}{8}, -\frac{1}{2}, -\frac{7}{16}, \right. \\ \left. -\frac{3}{8}, -\frac{5}{16}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{5}{16}, \frac{3}{8}, \frac{7}{16}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1, \frac{5}{4}, \frac{3}{2}, \frac{7}{4} \right\}$$

Rozložení všech čísel z  $S_2$  na číselné ose.



Všimněte si velké díry okolo čísla 0. Tam vzniká jev zvaný Underflow (podtečení).

Počet všech těchto čísel včetně nuly je

$$2 * 2 * 2 * 3 (*tři hodnoty exponentu*) + 1$$

25

Se symboly  $\pm \infty$  a NaN je jich celkem 28 různých.

Subnormální čísla (včetně  $\pm 0$ ):

$$\left\{ -\frac{3}{16}, -\frac{1}{8}, -\frac{1}{16}, 0, 0, \frac{1}{16}, \frac{1}{8}, \frac{3}{16} \right\} \\ \left\{ -\frac{7}{4}, -\frac{3}{2}, -\frac{5}{4}, -1, -\frac{7}{8}, -\frac{3}{4}, -\frac{5}{8}, -\frac{1}{2}, -\frac{7}{16}, -\frac{3}{8}, -\frac{5}{16}, -\frac{1}{4}, -\frac{3}{16}, \right. \\ \left. -\frac{1}{8}, -\frac{1}{16}, 0, \frac{1}{16}, \frac{1}{8}, \frac{3}{16}, \frac{1}{4}, \frac{5}{16}, \frac{3}{8}, \frac{7}{16}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1, \frac{5}{4}, \frac{3}{2}, \frac{7}{4} \right\}$$

Rozložení všech čísel včetně subnormálních:



Double precision IEEE 754 standart. Tento systém je tvořen konečnou podmnožinou z racionálních čísel (která mají simulovat nespočtenou množinu reálných čísel v počítači), které mohou být zapsány ve tvaru

$$x = \pm(1 + m_1 2^{-1} + m_2 2^{-2} + \dots + m_{52} 2^{-52}) \cdot 2^E, \quad (7)$$

kde  $m_i \in \{0, 1\}$  pro  $i = 1, 2, \dots, 52$ ,  
 $E = -1022, -1021, \dots, 0, \dots, 1022, 1023$ . Navíc obsahuje číslo 0 a symboly  $-\infty$ ,  $+\infty$  a NaN (not a number). Označme symbolem  $\mathbb{F}$  množinu všech racionálních čísel tohoto systému.

Číslo  $1 + 2^{-53}$  již není možné reprezentovat tímto systémem. Výsledná jeho hodnota je dána zaokrouhlením k číslu z množiny  $\mathbb{F}$ .

Double precision IEEE 754 standart.

$$x = \pm(1 + m_1 2^{-1} + m_2 2^{-2} + \dots + m_{52} 2^{-52}) \cdot 2^E, \quad (8)$$

kde  $m_i \in \{0, 1\}$  pro  $i = 1, 2, \dots, 52$ ,  
 $E = -1022, -1021, \dots, 0, \dots, 1022, 1023$ . Navíc obsahuje číslo 0 a symboly  $-\infty$ ,  $+\infty$  a NaN (not a number). Označme symbolem  $\mathbb{F}$  množinu všech racionálních čísel tohoto systému.

- $1 + 2^{-53}$  vrátí jako výsledek 1 při zaokrouhlení v módu k nejbližšímu číslu (to nearest).
- $1 + 2^{-53}$  vrátí jako výsledek 1 při zaokrouhlení v módu k  $-\infty$ .
- $1 + 2^{-53}$  vrátí jako výsledek  $1 + 2^{-52}$  při zaokrouhlení v módu k  $+\infty$ .

## Intervalová aritmetika

Počátky intervalové aritmetiky

První myšlenky - Archimédes a odhad čísla  $\pi$

Numerická matematika:

[M. Warmus: Calculus of Approximations, Bulletin de L' Académie Polonaise Des Sciences, Vol. 4, No. 5, 1956]

[T. Sunaga: Theory of Interval Algebra and its Applications to Numerical Analysis, RAAG Memoirs, Vol. 2, 1958]

[R. Moore: mnoho technických zpráv pro Lockheed Martin Missiles Department, 1960++]

V intervalové aritmetice je každé reálné číslo  $x \in \mathbb{R}$  reprezentováno intervalem  $X = [\underline{X}, \overline{X}] \subset \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$ , kde  $\underline{X} \in \mathbb{F} \cup \{-\infty\}$ ,  $\overline{X} \in \mathbb{F} \cup \{+\infty\}$  a  $\underline{X} \leq \overline{X}$ .

$$\mathbb{I}_{\mathbb{F}} \stackrel{\text{def}}{=} \{[\underline{X}, \overline{X}] : \underline{X} \in \mathbb{F} \cup \{-\infty\}, \overline{X} \in \mathbb{F} \cup \{+\infty\} \text{ a } \underline{X} \leq \overline{X}\}.$$

V intervalové aritmetice je definováno intervalové rozšíření

$$\oplus, \ominus, \odot, \oslash : \mathbb{I}_{\mathbb{F}}^2 \rightarrow \mathbb{I}_{\mathbb{F}}$$

operací  $+$ ,  $-$ ,  $\cdot$ ,  $/$  tak, aby

$$\begin{aligned} \forall X, Y \in \mathbb{I}_{\mathbb{F}} \quad \forall x, y \in \mathbb{R} : x \in X, y \in Y \implies \\ x + y \in X \oplus Y, \quad x - y \in X \ominus Y, \quad xy \in X \odot Y, \\ 0 \notin Y \implies \frac{x}{y} \in X \oslash Y. \end{aligned}$$

Operace  $X \oslash Y$  není definována pokud  $0 \in Y$ . Jinými slovy  $\oplus$  je

intervalovým rozšířením  $+$  když pro všechna  $X, Y \in \mathbb{I}_{\mathbb{F}}$ ,

$$[\underline{X} + \underline{Y}, \overline{X} + \overline{Y}] \subset X \oplus Y$$

**Poznámka:** intervalové rozšíření operace není definováno jednoznačně, neboť máme-li intervalové rozšíření operace, jakákoliv intervalová funkce dávající jako výsledek pro každý vstup větší interval je také intervalové rozšíření dané operace.

**Příklad** (optimální rozšíření součtu):  $[1, 1] \oplus [2^{-53}, 2^{-53}] = [1, 1 + 2^{-52}]$  (Dolní mez zaokrouhlena dolů k 1 a horní mez zaokrouhlena nahoru k  $1 + 2^{-52}$ ).

**Příklad** (optimální rozšíření rozdílu):  $[1, 1] \ominus [2^{-53}, 2^{-53}] = [1 - 2^{-53}, 1 - 2^{-53}]$  (Dolní mez i horní mez jsou výsledky zapsatelné v IEEE 754 normě formátu double, výsledek je tedy jednobodový interval).

**Příklad** (optimální rozšíření rozdílu):  $[1, 1] \ominus [2^{-54}, 2^{-54}] = [1 - 2^{-53}, 1]$  (Dolní mez zaokrouhlena dolů k  $1 - 2^{-53}$ ,

což je nejbližší menší číslo v IEEE 754 normě formátu double a horní mez je zaokrouhlena nahoru k 1, což je nejbližší větší číslo v daném formátu).

## Intervalová aritmetika v sw *Mathematica*

Zadáme-li do sw *Mathematica* vstup

```
Interval[{1., 1.}]
```

interval je automaticky zvětšen tak, že dolní mez je číslo předcházející zadané mezi ve formátu IEEE 754 (double). To zjistíme konverzí typu tohoto čísla na typ racionální (dvojice čísel, jmenovatel):

```
SetPrecision[%, Infinity]
```

```
Interval[{ { 9 007 199 254 740 991, 4 503 599 627 370 497 } / { 9 007 199 254 740 992, 4 503 599 627 370 496 } }]
```

Zkusme nyní provést následující součet intervalů:

```
Interval[{1., 1.}] + Interval[{2.^(-53), 2.^(-53)}]
```

```
Interval[{1., 1.}]
```

Konverzí typu výsledku na typ racionální (dvojice čísel, jmenovatel) zjistíme, že dolní mez se nezměnila, zatímco horní se posunula.

```
SetPrecision[%, Infinity]
```

```
Interval[{ { 9 007 199 254 740 991, 4 503 599 627 370 499 } / { 9 007 199 254 740 992, 4 503 599 627 370 496 } }]
```

Matematicky přesný výsledek leží v tomto intervalu.

```
IntervalMemberQ[%, 1 + 2^(-53)]
```

```
True
```

Dolní mez je posunutá o  $2^{-53}$  směrem dolů oproti 1

$$\left(1 - \frac{9\,007\,199\,254\,740\,991}{9\,007\,199\,254\,740\,992}\right) * 2^{53}$$

```
1
```

Horní mez je posunutá o  $3 \times 2^{-52}$  směrem nahoru oproti 1

$$\left(\frac{4\,503\,599\,627\,370\,499}{4\,503\,599\,627\,370\,496} - 1\right) * 2^{52}$$

```
3
```

**Zdůvodnění:** sw *Mathematica* je psán v jazyce C tak, aby byl přenositelný mezi různými platformami. Důsledek je ten, že nemá přístup k zaokrouhlovacím režimům procesoru (což lze jedině na úrovni jazyka Assembler, který je již závislý na procesoru). Proto se garantovaných odhadů dosahuje tak, že se výsledek posune k nejbližšímu menšímu IEEE 754 (double) číslu u dolní meze a k nejbližšímu většímu číslu IEEE 754 (double) u horní meze. Výpočet tedy proběhl takto:

1. konverze vstupu `Interval[{1.,1.}]` na `Interval[{1. - 2.-53, 1. + 2.-52}]`
2. konverze vstupu `Interval[{2.-53, 2.-53}]` na `Interval[{(1. - 2.-53) × 2.-53, (1. + 2.-52) × 2.-53}]`
3. součet dolních a horních mezí se zaokrouhlením k nejbližšímu: `Interval[{1., 1. + 2.-52 + 2.-52}]`
4. posuv mezí k nejbližší menší a nejbližší větší IEEE 754 (double) hodnotě:  
`Interval[{1. - 2.-53, 1. + 2.-52 + 2.-52 + 2.-52}] = Interval[{1. - 2.-53, 1. + 3 × 2.-52}]`

### ■ Experiment s konstantním polem.

Vytvoříme pole o  $10^6$  prvcích s konstantní hodnotou  $1/10$ :

```
a = ConstantArray[0.1, {10^6}];
```

Toto pole sečteme :

```
Total[a] // FullForm
```

```
100000.00000133288`
```

Zde je chyba od matematicky přesné hodnoty :

```
Total[a] - 100 000
```

```
1.33288 × 10-6
```

Nyní to samé provedeme v intervalové aritmetice

```
aInt = ConstantArray[Interval[{0.1, 0.1}], {10^6}];
```

Výsledek je interval

```
Plus@@aInt // FullForm
```

```
Interval[List[99999.9999931388`, 100000.000009527`]]
```

obsahující správnou hodnotu

```
IntervalMemberQ[%, 10^5]
```

```
True
```

### ■ Experiment s funkcí sinus.

Výpočet se strojově podporovanými čísly :

```
Table[{i, TraditionalForm[HoldForm[Sin[10^j Pi]]] /. j -> i, Sin[10^i Pi], Sin[10.^i Pi]},
  {i, 1, 20}] // TableForm
```

1	$\sin(10^1 \pi)$	0	$-1.22465 \times 10^{-15}$
2	$\sin(10^2 \pi)$	0	$1.96439 \times 10^{-15}$
3	$\sin(10^3 \pi)$	0	$-3.21417 \times 10^{-13}$
4	$\sin(10^4 \pi)$	0	$-4.85682 \times 10^{-13}$
5	$\sin(10^5 \pi)$	0	$-3.39607 \times 10^{-11}$
6	$\sin(10^6 \pi)$	0	$-2.23191 \times 10^{-10}$
7	$\sin(10^7 \pi)$	0	$5.62056 \times 10^{-10}$
8	$\sin(10^8 \pi)$	0	$-3.90829 \times 10^{-8}$
9	$\sin(10^9 \pi)$	0	$-3.32014 \times 10^{-8}$
10	$\sin(10^{10} \pi)$	0	$-2.23936 \times 10^{-6}$
11	$\sin(10^{11} \pi)$	0	-0.0000147642
12	$\sin(10^{12} \pi)$	0	-0.000269713
13	$\sin(10^{13} \pi)$	0	-0.00269712
14	$\sin(10^{14} \pi)$	0	-0.011346
15	$\sin(10^{15} \pi)$	0	-0.236209
16	$\sin(10^{16} \pi)$	0	-0.375213
17	$\sin(10^{17} \pi)$	0	-0.84797
18	$\sin(10^{18} \pi)$	0	-0.641653
19	$\sin(10^{19} \pi)$	0	0.746337
20	$\sin(10^{20} \pi)$	0	-0.394071

Výpočet v intervalové aritmetice se strojově podporovanými čísly :

```

Table[{i, TraditionalForm[HoldForm[Sin[10^j Pi]]] /. j -> i,
      Sin[10^i Pi], Sin[Interval[{10.^i Pi, 10.^i Pi}]]}, {i, 1, 20}] // TableForm

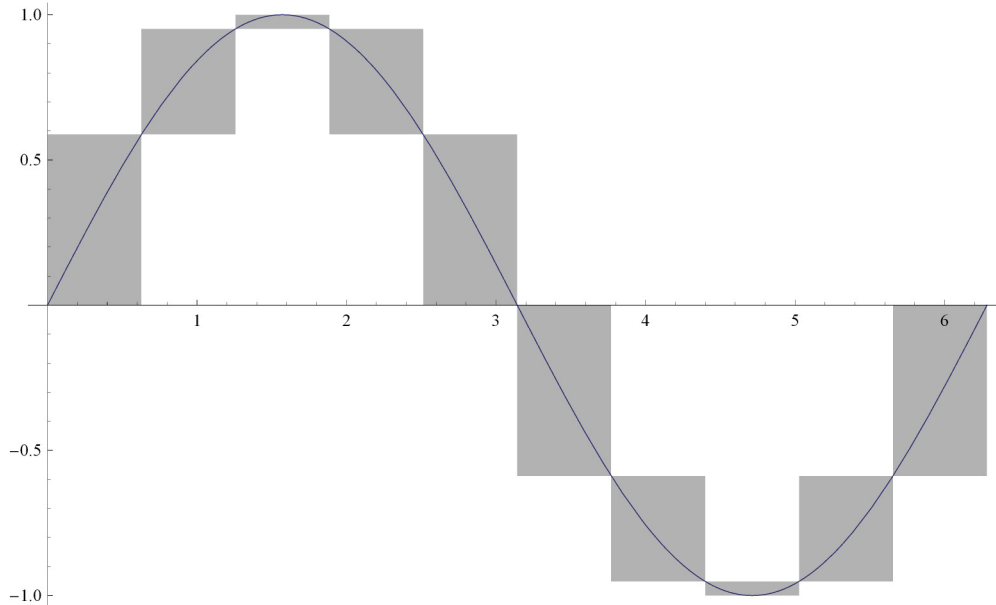
```

1	$\sin(10^1 \pi)$	0	Interval[{-4.77736 × 10 <sup>-15</sup> , 2.32807 × 10 <sup>-15</sup> }
2	$\sin(10^2 \pi)$	0	Interval[{-5.4879 × 10 <sup>-14</sup> , 5.88078 × 10 <sup>-14</sup> }
3	$\sin(10^3 \pi)$	0	Interval[{-7.76164 × 10 <sup>-13</sup> , 1.33331 × 10 <sup>-13</sup> }
4	$\sin(10^4 \pi)$	0	Interval[{-4.12366 × 10 <sup>-12</sup> , 3.1523 × 10 <sup>-12</sup> }
5	$\sin(10^5 \pi)$	0	Interval[{-9.21683 × 10 <sup>-11</sup> , 2.4247 × 10 <sup>-11</sup> }
6	$\sin(10^6 \pi)$	0	Interval[{-6.88853 × 10 <sup>-10</sup> , 2.4247 × 10 <sup>-10</sup> }
7	$\sin(10^7 \pi)$	0	Interval[{-3.16323 × 10 <sup>-9</sup> , 4.28735 × 10 <sup>-9</sup> }
8	$\sin(10^8 \pi)$	0	Interval[{-9.86876 × 10 <sup>-8</sup> , 2.05217 × 10 <sup>-8</sup> }
9	$\sin(10^9 \pi)$	0	Interval[{-5.10039 × 10 <sup>-7</sup> , 4.43636 × 10 <sup>-7</sup> }
10	$\sin(10^{10} \pi)$	0	Interval[{-6.05406 × 10 <sup>-6</sup> , 1.57533 × 10 <sup>-6</sup> }
11	$\sin(10^{11} \pi)$	0	Interval[{-0.0000757994, 0.0000462709}]
12	$\sin(10^{12} \pi)$	0	Interval[{-0.000757994, 0.000218569}]
13	$\sin(10^{13} \pi)$	0	Interval[{-0.00660333, 0.00120912}]
14	$\sin(10^{14} \pi)$	0	Interval[{-0.0737792, 0.0511314}]
15	$\sin(10^{15} \pi)$	0	Interval[{-0.673152, 0.258566}]
16	$\sin(10^{16} \pi)$	0	Interval[{-1, 1}]
17	$\sin(10^{17} \pi)$	0	Interval[{-1, 1}]
18	$\sin(10^{18} \pi)$	0	Interval[{-1, 1}]
19	$\sin(10^{19} \pi)$	0	Interval[{-1, 1}]
20	$\sin(10^{20} \pi)$	0	Interval[{-1, 1}]



## ■ Vykreslení jednoduchého intervalového grafu.

```
Plot[Sin[x], {x, 0, 2 Pi},
  Epilog -> Prepend[Table[{Interval[{2 Pi * i / 10, 2 Pi * (i + 1) / 10}], Sin[
    Interval[{2 Pi * i / 10, 2 Pi * (i + 1) / 10}]}], {i, 0., 9., 1.}] /.
  {Interval[{a_, b_}], Interval[{c_, d_}]} => Rectangle[{a, c}, {b, d}], Opacity[0.3]]
```



Vypadá to, že obdélníky v obrázku mají společné jen vrcholy, což je teoreticky možné jen v případě, že hodnota funkce sinus v daném bodě je strojově zobrazitelné číslo (v implementaci intervalové aritmetiky v sw *Mathematica* by se nepřekrývali ani v tomto případě). Následující výpočet ukáže, že překryv ve směru osy  $y$  je interval kladné šířky.

```
Partition[Table[Sin[Interval[{2 Pi * i / 10, 2 Pi * (i + 1) / 10}]]], {i, 0., 9., 1.}], 2, 1] /.
  {a_, b_} => IntervalIntersection[a, b] /. Interval[{a_, b_}] => b - a
```

```
{4.44089 × 10-16, 4.44089 × 10-16, 4.44089 × 10-16, 9.99201 × 10-16,
  8.88178 × 10-16, 8.88178 × 10-16, 7.77156 × 10-16, 8.88178 × 10-16, 1.66533 × 10-15}
```

Stejně tak to dopadne s překryvem ve směru osy  $x$ .

```
Partition[Table[Interval[{2 Pi * i / 10, 2 Pi * (i + 1) / 10}], {i, 0., 9., 1.}], 2, 1] /.
  {a_, b_} => IntervalIntersection[a, b] /. Interval[{a_, b_}] => b - a
```

```
{2.22045 × 10-16, 4.44089 × 10-16, 4.44089 × 10-16, 8.88178 × 10-16,
  8.88178 × 10-16, 8.88178 × 10-16, 1.77636 × 10-15, 1.77636 × 10-15, 1.77636 × 10-15}
```

**Závěr:** z výše uvedených hodnot je patrné, že se intervaly překrývají jak horizontálně, tak vertikálně. Překryv v řádu  $10^{-15}$  však v obrázku s hodnotami na osách v řádu  $10^0$  patrný není.

## ■ Jednoduchý intervalový integrátor.

Myšlenka je založena na Riemannových součtech.

$$\int_a^b f(x) dx \in \sum_{i=0}^{n-1} \left[ \min_{s \in [a+i(b-a)/n, a+(i+1)(b-a)/n]} f(s), \max_{s \in [a+i(b-a)/n, a+(i+1)(b-a)/n]} f(s) \right] \otimes [(b-a)/n, (b-a)/n],$$

kde dolní meze v intervalech  $[a + i(b-a)/n, a + (i+1)(b-a)/n]$  a  $[(b-a)/n, (b-a)/n]$  jsou dolní strojově zobrazitelné odhady přesných hodnot  $a + i(b-a)/n$  a  $(b-a)/n$ . Horní meze těchto intervalů jsou horní strojově

zobrazitelné odhady přesných hodnot  $a + (i + 1)(b - a)/n$  a  $(b - a)/n$ . Sumu  $\sum_{i=1}^n$  chápeme jako sumu  $n$  intervalů operátorem  $\oplus$ .

Integrál  $\int_0^1 \sin(x) dx$  snadno vypočtemem analyticky a použijeme jej k testu našeho intervalového integrátoru.

```
Integrate[Sin[x], {x, 0, 1}]
```

```
1 - Cos[1]
```

Tato přesná hodnota se nachází v intervalu:

```
intAnalytic = N[Interval[%, %]]
```

```
Interval[{0.459698, 0.459698}]
```

Numerickou metodou (numerickou kvadraturou) počítající pouze se strojově zobrazitelnými čísly dostaneme :

```
numericIEEEDouble = NIntegrate[Sin[x], {x, 0, 1}]
```

```
0.459698
```

```
IntervalMemberQ[intAnalytic, numericIEEEDouble]
```

```
False
```

Vidíme že hodnota vypočtená numerickou kvadraturou leží mimo dolní a horní odhad analytické hodnoty. Zde je odhad chyby od přesné hodnoty.

```
Integrate[Sin[x], {x, 0, 1}] - NIntegrate[Sin[x], {x, 0, 1}]
```

```
- 4.44089 × 10-16
```

Nyní provedeme numerickou integraci v intervalové aritmetice.

```
intervalRiemannSum = Total[Sin[#] * Interval[{1 / 1000., 1 / 1000.}] & /@
```

```
Table[Interval[{i / 1000, (i + 1) / 1000}], {i, 0., 999.}]]
```

```
Interval[{0.459277, 0.460118}]
```

```
IntervalMemberQ[%, intAnalytic]
```

```
True
```

**Závěr:** Výsledek získaný intervalovým integrátorem obsahuje správný výsledek.

**Pozn.:** Příkaz `IntervalMemberQ[interval1, interval2]` vrací hodnotu `True` tehdy a jen tehdy, je-li *interval*<sub>2</sub> podmnožinou *interval*<sub>1</sub>

## Počítačem asistovaný důkaz existence/neexistence řešení

### ■ Fredholmova alternativa

Známým faktem je (viz [Drábek, Pavel; Milota, Jaroslav Methods of nonlinear analysis. Applications to differential equations. Birkhäuser Advanced Texts: Basler Lehrbücher. [Birkhäuser Advanced Texts: Basel Textbooks] Birkhäuser Verlag, Basel, 2007. xii+568 pp. ISBN: 978-3-7643-8146-2, S. Fučík: Solvability of nonlinear equations and boundary value problems. Mathematics and its Applications, 4. D. Reidel Publishing Co., Dordrecht-Boston, Mass., 1980. vii+390 pp. ISBN: 90-277-1077-5], že nutná a postačující podmínka pro existenci řešení okrajové úlohy

$$-u''(x) - u(x) = f(x), \quad u(0) = 0 = u(\pi)$$

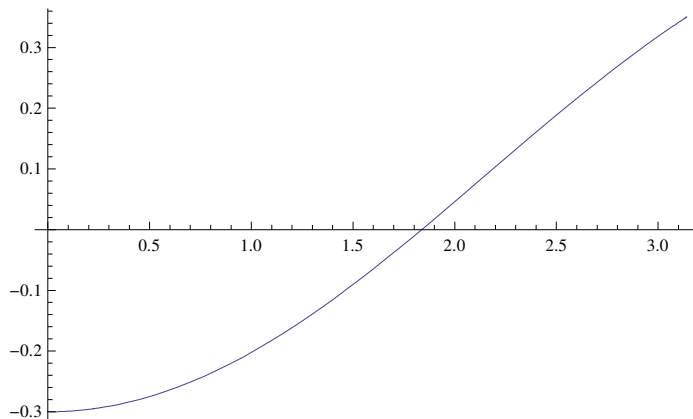
má tvar

$$\int_0^\pi f(x) \sin(x) dx = 0.$$

Výpočtem v intervalové aritmetice sice nemůžeme dokázat rovnost, ale můžeme dokázat nerovnost, t.j. v našem případě neexistence řešení.

Nechť např.  $f(x) = \sin(\sin(x/\pi)^2) - 3/10$ . Máme zjistit, zda pro tuto funkci má výše uvedená rovnice řešení.

```
Plot[Sin[Sin[x / Pi]^2] - 3 / 10, {x, 0, Pi}]
```



Tento integrál lze analyticky vypočítat jen stěží.

```
Integrate[(Sin[Sin[x / Pi]^2] - 3 / 10) Sin[x], {x, 0, Pi}]
```

$$\int_0^\pi \sin[x] \left( -\frac{3}{10} + \sin\left[\sin\left[\frac{x}{\pi}\right]^2\right] \right) dx$$

Numerickou kvadraturou jej vypočítat lze, ale nevíme nic o tom, jak daleko je od přesné hodnoty.

```
NIntegrate[(Sin[Sin[x / Pi]^2] - 3 / 10) Sin[x], {x, 0, 1}]
```

```
-0.115804
```

Intervalová integrace zaručuje, že přesný výsledek leží ve výsledném intervalu.

```
Total[(Sin[Sin[# / Pi]^2] - 3 / 10) Sin[#] * Interval[{1 / 1000., 1 / 1000.}] & /@
```

```
Table[Interval[{i / 1000, (i + 1) / 1000}], {i, 0., 999.}]]
```

```
Interval[{-0.115948, -0.115661}]
```

Tedy  $\int_0^\pi (\sin(\sin(x/\pi)^2) - 3/10) \sin(x) dx \in [-0.115948, -0.115661] \subset [0, 0]$ .

**Závěr:** výše uvedená úloha řešení nemá.

■ **Landesmanova - Lazerova podmínka :**

Nutná i postačující podmínka pro existenci řešení rovnice

$$-u''(x) - u(x) = \arctg(u(x)) + f(x), \quad u(0) = 0 = u(\pi)$$

je

$$-\pi < \int_0^\pi f(x) \sin(x) dx < \pi,$$

viz [Landesman, E. M.; Lazer, A. C. Nonlinear perturbations of linear elliptic boundary value problems at resonance. J. Math. Mech. 19 1969/1970 609–623. S. Fučík: Solvability of nonlinear equations and boundary value problems. Mathematics and its Applications, 4. D. Reidel Publishing Co., Dordrecht-Boston, Mass., 1980. vii+390 pp. ISBN: 90-277-1077-5]

Výpočtem v intervalové aritmetice sice nemůžeme dokázat rovnost, ale můžeme dokázat nerovnost, t.j. v našem případě existenci i neexistenci řešení.

Nechť  $f(x) = \sin(\sin(x/\pi)^2) - 3/10$ . Podle výše uvedeného výpočtu  $\int_0^\pi (\sin(\sin(x/\pi)^2) - 3/10) \sin(x) dx \in [-0.115948, -0.115661] \subset (-\pi, \pi)$ .

**Závěr:** výše uvedená úloha řešení má.

## Intervalové integrování počáteční úlohy

V praxi se často vyskytují počáteční úlohy

$$y'(x) = f(x, y(x)), \quad y(0) = y_0$$

jejichž řešení nelze přesně určit analyticky. Takové úlohy se řeší numericky. Někdy potřebujeme mít garantovaný odhad, kde se hodnoty řešení nalézají. Například v matematických důkazech. Toho nelze docílit standartními numerickými metodami integrace počátečních úloh a musíme použít intervalové metody integrace.

### Intervalová integrace počáteční úlohy

Diferenciální rovnici

$$y'(x) = f(x, y(x))$$

převědeme integraci od  $x_n$  do  $x_{n+1}$  na integrální

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x)) dx.$$

#### Tvrzení:

Nechť  $y(x_n) \in Y_n$ ,  $X_{n,n+1} = [x_n, x_{n+1}]$ ,  $x_{n+1} - x_n \in H_n$ ,  $\sup H_n \geq \inf H_n > 0$ , a interval  $Y_{n,n+1}$  je takový, že

$$\forall x \in X_{n,n+1} : y(x) \in Y_{n,n+1},$$

pak

$$y(x_{n+1}) \in Y_{n+1} := Y_n \oplus H_n \otimes \left[ \inf_{x \in X_{n,n+1}, y \in Y_{n,n+1}} f(x, y), \sup_{x \in X_{n,n+1}, y \in Y_{n,n+1}} f(x, y) \right].$$

Navíc platí:

$$\forall x \in X_{n,n+1} : y(x) \in \text{Hull}(Y_n, Y_{n+1}) \subset Y_{n,n+1}.$$

**Důkaz:** Dle předpokladu  $y(x_n) \in Y_n$ . Dále podle monotonie integrálu (podle meze i podle integrandu) dostaneme

$$\int_{x_n}^{x_{n+1}} f(x, y(x)) dx \in H_n \otimes \left[ \inf_{x \in X_{n,n+1}, y \in Y_{n,n+1}} f(x, y), \sup_{x \in X_{n,n+1}, y \in Y_{n,n+1}} f(x, y) \right].$$

Z definice operace  $\oplus$  pak již dostaneme tuto inkluzi:

$$y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x)) dx \in Y_n \oplus H_n \otimes \left[ \inf_{x \in X_{n,n+1}, y \in Y_{n,n+1}} f(x, y), \sup_{x \in X_{n,n+1}, y \in Y_{n,n+1}} f(x, y) \right]$$

Z rovnosti  $y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x)) dx$ , pak již plyne dokazovaná inkluze.

Zbývá ukázat  $\forall x \in X_{n,n+1} : y(x) \in \text{Hull}(Y_n, Y_{n+1}) \subset Y_{n,n+1}$ .

Nechť  $x \in X_{n,n+1}$ , potom integrací rovnice dostaneme  $y(x) = y(x_n) + \int_{x_n}^x f(x, y(x)) dx$ . Dále se využívá monotonie integrálu

a nerovnost

$$\sup H_n \geq \inf H_n > 0:$$

$$y(x) \geq \inf Y_n + \inf_{x \in X_{n,n+1}, y \in Y_{n,n+1}} \int_{x_n}^x f(x, y) dx \geq \inf Y_n + \min(0, \int_{x_n}^{x_{n+1}} \inf_{x \in X_{n,n+1}, y \in Y_{n,n+1}} f(x, y) dx) \geq$$

$$\inf Y_n + \min(0, \sup H_n \cdot \inf_{x \in X_{n,n+1}, y \in Y_{n,n+1}} f(x, y) dx) \geq \min(\inf Y_n, \inf Y_{n+1}) = \inf \text{Hull}(Y_n, Y_{n+1}).$$

Stejně bychom dokázali nerovnost  $y(x) \leq \sup \text{Hull}(Y_n, Y_{n+1})$ . Konec důkazu.

**Algoritmus integrace počáteční úlohy.**  $m$  integračních kroků.

$Y_1$  zvol tak, aby  $y(0) \in Y_1$

**Pro  $n$  od 1 do  $m$  proved'**

$Y_{n+1} \leftarrow$  Počáteční garantovaný odhad  $Y_{n+1}$  (např. Gronwallovo Lemma, energetické nerovnosti apod.);

**Opakuj  $k$  – krát :**

$$Y_{n+1} \leftarrow Y_n \oplus H_n \otimes \left[ \inf_{x \in X_{n,n+1}, y \in \text{Hull}(Y_n, Y_{n+1})} f(x, y), \sup_{x \in X_{n,n+1}, y \in \text{Hull}(Y_n, Y_{n+1})} f(x, y) \right];$$

Konec **Opakuj** ;

$Y_n \leftarrow Y_{n+1}$ ;

$n \leftarrow n + 1$ ;

Konec **Pro ... proved'**

```
Hull[Interval[{a_, b_}], Interval[{c_, d_}]] := Interval[{Min[a, b], Max[c, d]}]
```

```
Hull[Interval[{0, 1}], Interval[{2, 3}]]
```

```
Interval[{0, 3}]
```

Zde je jednoduchý program, který provede intervalovou integraci obyčejné diferenciální rovnice

$$y'(x) = \arctg(y(x)), \quad y(0) = 1.$$

Tento program je založený na přepisu počáteční úlohy na integrální rovnici a v podstatě se jedná o intervalovou analogii Eulerovy implicitní metody.

```
Clear[h, Y];
```

```
h = 0.01;
```

```
Y = Interval[{1., 1.}];
```

```
vysl = Reap[For[
```

```
  n = 0, n < 100, n++,
```

```
  Ypom = Interval[{-100, 100}];
```

```
  For[i = 0, i < 30, i++, Ypom = Y + ArcTan[Hull[Y, Ypom]] * Interval[{h, h}]
```

```
  ];
```

```
  Y = Ypom;
```

```
  Sow[Y]
```

```
]]
```

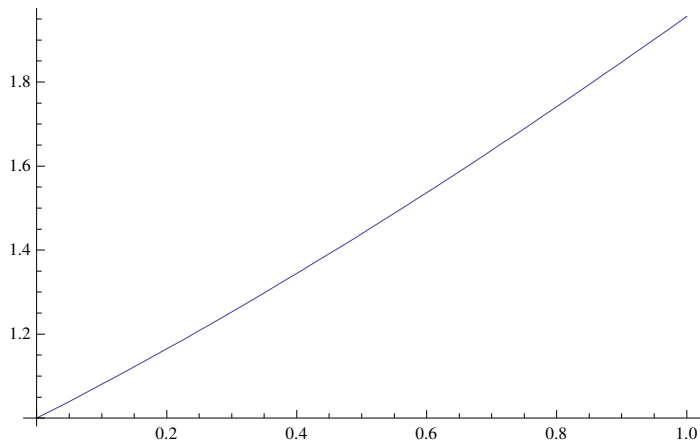
## ■ Příklad

Ukážeme si to na příkladu. Dáváme v něm přednost jednoduchosti před praktičností.

$$y'(x) = \arctg(y(x)), \quad y(0) = 1$$

Řešení numerickou metodou bez odhadu chyby.

```
gr = Plot[Evaluate[
  (y[t] /. NDSolve[{y'[x] == ArcTan[y[x]], y[0] == 1}, y, {x, 0, 1}][[1]]), {t, 0, 1}]
```



Pro intervalovou integraci potřebujeme definovat konvexní obal dvou intervalů.

```
Hull[Interval[{a_, b_}], Interval[{c_, d_}]] := Interval[{Min[a, b], Max[c, d]}]
```

```
Hull[Interval[{0, 1}], Interval[{2, 3}]]
```

```
Interval[{0, 3}]
```

Zde je jednoduchý program, který provede intervalovou integraci obyčejné diferenciální rovnice. Tento program je založený na přepisu počáteční úlohy na integrální rovnici a v podstatě se jedná o intervalovou analogii Eulerovy implicitní metody.

```
Clear[h, Y];
```

```
h = 0.01;
```

```
Y = Interval[{1., 1.}];
```

```
vysl = Reap[For[
```

```
  n = 0, n < 100, n++,
```

```
  Ypom = Interval[{-100, 100}];
```

```
  For[i = 0, i < 30, i++, Ypom = Y + ArcTan[Hull[Y, Ypom]] * Interval[{h, h}]
```

```
  ];
```

```
  Y = Ypom;
```

```
  Sow[Y]
```

```
]]
```

```
{Null, {{Interval[{1.00785, 1.00789}], Interval[{1.01575, 1.01583}],
  Interval[{1.02368, 1.0238}], Interval[{1.03165, 1.03181}], Interval[{1.03966, 1.03986}],
  Interval[{1.04771, 1.04795}], Interval[{1.0558, 1.05607}], Interval[{1.06392, 1.06424}],
  Interval[{1.07208, 1.07244}], Interval[{1.08029, 1.08068}],
  Interval[{1.08853, 1.08896}], Interval[{1.0968, 1.09728}], Interval[{1.10512, 1.10563}],
  Interval[{1.11347, 1.11403}], Interval[{1.12186, 1.12246}],
  Interval[{1.13029, 1.13092}], Interval[{1.13875, 1.13943}],
  Interval[{1.14726, 1.14797}], Interval[{1.15579, 1.15655}],
  Interval[{1.16437, 1.16517}], Interval[{1.17298, 1.17382}],
  Interval[{1.18163, 1.18251}], Interval[{1.19032, 1.19123}],
  Interval[{1.19904, 1.19999}], Interval[{1.20779, 1.20879}],
  Interval[{1.21658, 1.21762}], Interval[{1.22541, 1.22649}],
  Interval[{1.23428, 1.23539}], Interval[{1.24318, 1.24433}],
  Interval[{1.25211, 1.2533}], Interval[{1.26108, 1.26231}], Interval[{1.27008, 1.27135}],
  Interval[{1.27912, 1.28043}], Interval[{1.28819, 1.28954}],
  Interval[{1.2973, 1.29869}], Interval[{1.30644, 1.30787}], Interval[{1.31561, 1.31708}],
  Interval[{1.32482, 1.32633}], Interval[{1.33407, 1.33561}],
  Interval[{1.34334, 1.34493}], Interval[{1.35265, 1.35427}],
  Interval[{1.36199, 1.36365}], Interval[{1.37137, 1.37307}],
  Interval[{1.38077, 1.38251}], Interval[{1.39021, 1.39199}],
  Interval[{1.39969, 1.4015}], Interval[{1.40919, 1.41105}], Interval[{1.41873, 1.42062}],
  Interval[{1.42829, 1.43023}], Interval[{1.43789, 1.43986}],
  Interval[{1.44753, 1.44953}], Interval[{1.45719, 1.45923}],
  Interval[{1.46688, 1.46896}], Interval[{1.47661, 1.47873}],
  Interval[{1.48636, 1.48852}], Interval[{1.49615, 1.49834}],
  Interval[{1.50596, 1.50819}], Interval[{1.51581, 1.51808}],
  Interval[{1.52569, 1.52799}], Interval[{1.53559, 1.53793}], Interval[{1.54553, 1.5479}],
  Interval[{1.55549, 1.55791}], Interval[{1.56549, 1.56794}], Interval[{1.57551, 1.578}],
  Interval[{1.58556, 1.58808}], Interval[{1.59564, 1.5982}], Interval[{1.60575, 1.60835}],
  Interval[{1.61589, 1.61852}], Interval[{1.62606, 1.62872}],
  Interval[{1.63625, 1.63895}], Interval[{1.64647, 1.64921}],
  Interval[{1.65672, 1.65949}], Interval[{1.667, 1.66981}], Interval[{1.67731, 1.68014}],
  Interval[{1.68764, 1.69051}], Interval[{1.698, 1.7009}], Interval[{1.70838, 1.71132}],
  Interval[{1.71879, 1.72177}], Interval[{1.72923, 1.73224}],
  Interval[{1.7397, 1.74274}], Interval[{1.75019, 1.75327}], Interval[{1.76071, 1.76382}],
  Interval[{1.77125, 1.77439}], Interval[{1.78182, 1.78499}],
  Interval[{1.79241, 1.79562}], Interval[{1.80303, 1.80627}],
  Interval[{1.81367, 1.81695}], Interval[{1.82434, 1.82765}],
  Interval[{1.83504, 1.83838}], Interval[{1.84576, 1.84913}], Interval[{1.8565, 1.8599}],
  Interval[{1.86727, 1.8707}], Interval[{1.87806, 1.88152}], Interval[{1.88887, 1.89237}],
  Interval[{1.89971, 1.90324}], Interval[{1.91057, 1.91413}],
  Interval[{1.92146, 1.92505}], Interval[{1.93237, 1.93599}],
  Interval[{1.9433, 1.94695}], Interval[{1.95426, 1.95794}]}}
```

Nyní se pokusíme o grafický výstup těchto intervalů.

Zde do výsledku přidáme počáteční podmínku, vytvoříme dvojice intervalových hodnot  $Y_i$  a  $Y_{i+1}$  v předchozím a následujícím kroku a vytvoříme z této dvojice konvexní obal. Hodnoty řešení  $y(x)$  pro  $x \in [x_i, x_{i+1}]$  leží v tomto konvexním obalu. Nakonec ještě Intervaly převedeme na uspořádanou dvojici čísel {horní mez, dolní mez}.

```
ints = (Hull@@# &) /@ Partition[Prepend[vysl[[2, 1]], Interval[{1., 1.}], 2, 1] /.
  Interval[a__] -> a;
```

Ještě vytvoříme intervaly pro nezávisle proměnnou  $x$ . Nakonec opět Intervaly převedeme na uspořádanou dvojici čísel {horní mez, dolní mez}.

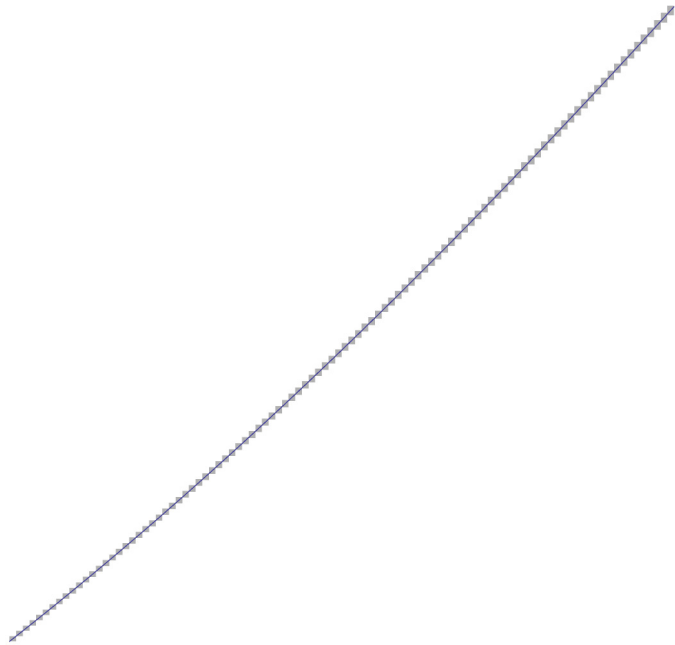


```

hs = (Table[Interval[{i, i + 1}] * 0.01, {i, 0, 99}]) /. Interval[a_] => a
{{-2.22507 × 10-308, 0.01}, {0.01, 0.02}, {0.02, 0.03}, {0.03, 0.04},
 {0.04, 0.05}, {0.05, 0.06}, {0.06, 0.07}, {0.07, 0.08}, {0.08, 0.09}, {0.09, 0.1},
 {0.1, 0.11}, {0.11, 0.12}, {0.12, 0.13}, {0.13, 0.14}, {0.14, 0.15}, {0.15, 0.16},
 {0.16, 0.17}, {0.17, 0.18}, {0.18, 0.19}, {0.19, 0.2}, {0.2, 0.21}, {0.21, 0.22},
 {0.22, 0.23}, {0.23, 0.24}, {0.24, 0.25}, {0.25, 0.26}, {0.26, 0.27}, {0.27, 0.28},
 {0.28, 0.29}, {0.29, 0.3}, {0.3, 0.31}, {0.31, 0.32}, {0.32, 0.33}, {0.33, 0.34},
 {0.34, 0.35}, {0.35, 0.36}, {0.36, 0.37}, {0.37, 0.38}, {0.38, 0.39}, {0.39, 0.4},
 {0.4, 0.41}, {0.41, 0.42}, {0.42, 0.43}, {0.43, 0.44}, {0.44, 0.45}, {0.45, 0.46},
 {0.46, 0.47}, {0.47, 0.48}, {0.48, 0.49}, {0.49, 0.5}, {0.5, 0.51}, {0.51, 0.52},
 {0.52, 0.53}, {0.53, 0.54}, {0.54, 0.55}, {0.55, 0.56}, {0.56, 0.57}, {0.57, 0.58},
 {0.58, 0.59}, {0.59, 0.6}, {0.6, 0.61}, {0.61, 0.62}, {0.62, 0.63}, {0.63, 0.64},
 {0.64, 0.65}, {0.65, 0.66}, {0.66, 0.67}, {0.67, 0.68}, {0.68, 0.69}, {0.69, 0.7},
 {0.7, 0.71}, {0.71, 0.72}, {0.72, 0.73}, {0.73, 0.74}, {0.74, 0.75}, {0.75, 0.76},
 {0.76, 0.77}, {0.77, 0.78}, {0.78, 0.79}, {0.79, 0.8}, {0.8, 0.81}, {0.81, 0.82},
 {0.82, 0.83}, {0.83, 0.84}, {0.84, 0.85}, {0.85, 0.86}, {0.86, 0.87}, {0.87, 0.88},
 {0.88, 0.89}, {0.89, 0.9}, {0.9, 0.91}, {0.91, 0.92}, {0.92, 0.93}, {0.93, 0.94},
 {0.94, 0.95}, {0.95, 0.96}, {0.96, 0.97}, {0.97, 0.98}, {0.98, 0.99}, {0.99, 1.}}

Show[Graphics[Prepend[({ints, hs} // Transpose) /.
  {{a_?NumberQ, b_}, {c_, d_}} => Rectangle[{c, a}, {d, b}], Opacity[0.3]], gr]

```



Podle obrázku to vypadá, že se intervaly nepřekrývají. Dá se však matematicky rigorózně dokázat, že se překrývají.

Pro ilustraci si zkusme vypsát a vykreslit první tři intervaly. Zde je překryv vidět.

```

ints[[1 ;; 3]]
{{1., 1.00789}, {1.00785, 1.01583}, {1.01575, 1.0238}}

```

```
Graphics[Prepend[({ints[[1 ;; 3]], hs[[1 ;; 3]]} // Transpose) /.
  {{a_?NumberQ, b_}, {c_, d_}} => Rectangle[{c, a}, {d, b}], Opacity[0.3]]]
```



Šíře překryvu intervalů se během výpočtu zvětšuje.

Pro ilustraci si zkusme vypsát a vykreslit poslední tři intervaly.

```
ints[[-3 ;; -1]]
```

```
{1.92146, 1.93599}, {1.93237, 1.94695}, {1.9433, 1.95794}
```

```
Graphics[Prepend[({ints[[-3 ;; -1]], hs[[-3 ;; -1]]} // Transpose) /.
```

```
  {{a_?NumberQ, b_}, {c_, d_}} => Rectangle[{c, a}, {d, b}], Opacity[0.3]]]
```



---

## Závěr

Čtenář byl seznámen se zdrojem chyb ve vědecko-technických výpočtech na moderních počítačích. V textu jsme vysvětlili základní principy intervalové aritmetiky a možnosti jejího využití v matematických důkazech. Jako příklad jsme použili jednoduchý integrátor založený na Riemannových součtech a jednoduchý integrátor obyčejné diferenciální rovnice založený a intervalové modifikaci Eulerovy metody.