Tomáš Oberhuber

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

Video na Youtube

Reprezentace čísel s pohyblivou desetinnou čárkou

Definition 1

Buď základ $\beta \in \mathbb{N}$ pevně dané číslo $\beta \geq 2$, x buď reálné číslo s konečným počtem cifer $0 \leq x_k < \beta$ pro $k = -m, \ldots, n$. Pak **poziční zápis čísla v soustavě se základem** β má tvar

$$x_\beta = (-1)^s \left[x_n x_{n-1} \ldots x_1 x_0 . x_{-1} x_{-2} \ldots x_{-m} \right],$$

kde požadujeme, aby platilo $x_n \neq 0$. s udává znaménko čísla x ($s = 0$, je-li x kladné a $s = 1$, je-li x záporné). Také lze psát

$$x_\beta = (-1)^s \left(\sum_{k=-m}^{n} x_k \beta^k \right).$$

Reprezentace čísel s pohyblivou desetinnou čárkou

Theorem 2

Libovolné reálné číslo $x \in \mathbb{R}$ lze s libovolnou přesností aproximovat reálným číslem x_β, jehož zápis v soustavě o základě β má konečný počet cifer.

Důkaz.

Stačí jen počet cifer volit dostatečně velký. □

Reprezentace čísel s pohyblivou desetinnou čárkou

Remark 3

Při ukládání hodně velkých nebo hodně malých čísel v počítači nás ale uvedený způsob zápisu nutí ukládat do paměti velký počet cifer:

- $x = 0.00000000000000012348283$ - *velký počet nul*

Proto zavádíme exponenty a omezujeme maximální počet cifer zápisu daného čísla.

Reprezentace čísel s pohyblivou desetinnou čárkou

Definition 4

Buď $\beta \in \mathbb{N}, \beta > 1$ základ číselného rozvoje, $t \in \mathbb{N}$ **maximální počet povolených významných cifer** $0 \leq a_i < \beta$ pro $i \in \hat{t}$, $e \in \mathbb{N}$ exponent a s parametr vyjadřující znaménko daného čísla. Pak x zapisujeme ve tvaru

$$x = (-1)^s \cdot (0.a_1 a_2 \dots a_t) \cdot \beta^e = (-1)^s \cdot m \cdot \beta^{e-t},$$

kde $m = a_1 a_2 \dots a_t$ je **mantisa** čísla x.

Reprezentace čísel s pohyblivou desetinnou čárkou

Remark 5

Pro jednoznačnost zápisu požadujeme, aby první cifra mantisy byla nenulová, tj. $a_1 \neq 0$, tudíž $m \geq \beta^{t-1}$. Jinak by např. platilo

$$1 = 0.100 \cdot 10^1 = 0.010 \cdot 10^2 = 0.001 \cdot 10^3.$$

Definition 6

Zápisu čísla, který splňuje podmínku, že $a_1 \neq 0$, se říká **normalizovaný**.

Reprezentace čísel s pohyblivou desetinnou čárkou

Remark 7

Všimněme si, že normalizovaný zápis čísla **neumožňuje** *zapsat číslo 0! Nula bývá definována speciálně - to uvidíme později.*

Definition 8

Definujeme **množinu čísel s pohyblivou desetinou čárkou** jako

$$\mathbb{F}(\beta, t, L, U) \equiv \{0\} \cup \left\{ x \in \mathbb{R} \mid x = (-1)^s \beta^e \sum_{i=1}^{t} a_i \beta^{-i} \right\},$$

kde $\beta \in \mathbb{N}, \beta > 1$ určuje základ rozvoje $t \in \mathbb{N}, t > 0$ určuje počet významných cifer, a pro exponent $e \in \mathbb{N}$ platí $L \leq e \leq U$.

Reprezentace čísel s pohyblivou desetinnou čárkou

Remark 9

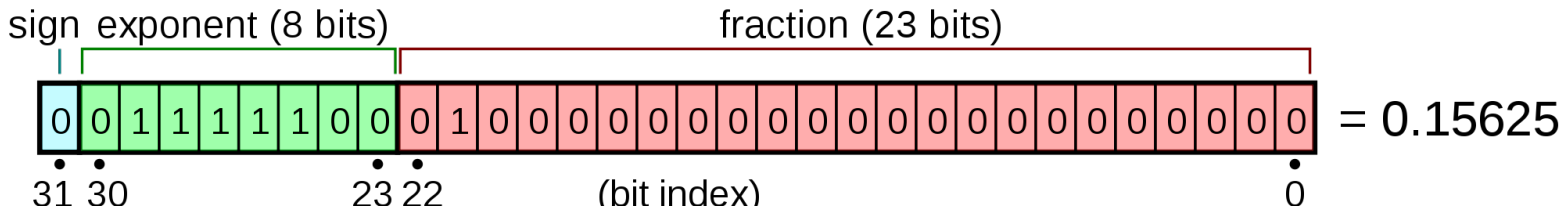
- *při práci s počítačem nejčastěji používáme základy* $\beta = 2, 10, 16$
- *interně počítače pracují nejčastěji se základy* 2, *výjimečně i* 10
- *při binárním tj.* $\beta = 2$ *zpracování čísel s pohyblivou desetinou čárkou je potřeba (je-li k dispozici N bitů)*
  - 1 *bit pro uložení znaménka s*
  - *t bitů pro uložení mantisy m*
  - $N - t - 1$ *bitů pro uložení exponentu e*

Reprezentace čísel s pohyblivou desetinnou čárkou

Standard IEEE (Institute of Electrictal and Electronics Engineers) definuje **jednoduchou a dvojitou přesnost** (single and double precision)

Reprezentace čísel s pohyblivou desetinnou čárkou

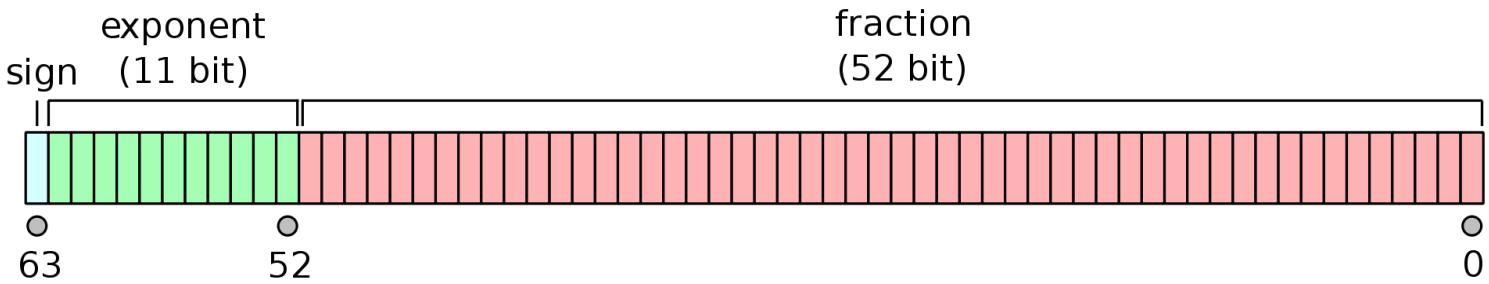
Single precision (32 bitů = 4 bajty)



- 1 bit pro znaménko
- 8 bitů pro exponent ($L = -126, U = 127$)
- 23 bitů pro mantisu (0–8 388 608)

Reprezentace čísel s pohyblivou desetinnou čárkou

Double precision (64 bitů = 8 bajtů)



- 1 bit pro znaménko
- 11 bitů pro exponent ($L = -1\,022$, $U = 1\,023$)
- 52 bitů pro mantisu (0–4 503 599 627 370 496)

Reprezentace čísel s pohyblivou desetinnou čárkou

Standard IEEE dále definuje denormalizovaná čísla:

hodnota	exponent	mantisa
± 0	$L-1$	0
$\pm\infty$	$U+1$	0
NaN	$U+1$	$\neq 0$

NaN znamená - **Not a Number**. Nastává při nedefinovaných operacích, např.

- dělení nulou
- odmocnina nebo logaritmus ze záporného čísla
- ...

Reprezentace čísel s pohyblivou desetinnou čárkou

Jaké je rozložení množiny $\mathbb{F}(\beta, t, L, U)$ v $\mathbb{R}$?

Example 10

Buď $t = 1$, $\beta = 10$, $L = -2$, $U = 2$.$\mathbb{F}(\beta, t, L, U)$ obsahuje nulu a následující čísla [1]:

	$e = -2$	$e = -1$	$e = 0$	$e = 1$	$e = 2$
$a_1 = 1$	0.01	0.1	1	10	100
$a_1 = 2$	0.02	0.2	2	20	200
$a_1 = 3$	0.03	0.3	3	30	300
$a_1 = 4$	0.04	0.4	4	40	400
$a_1 = 5$	0.05	0.5	5	50	500
$a_1 = 6$	0.06	0.6	6	60	600
$a_1 = 7$	0.07	0.7	7	70	700
$a_1 = 8$	0.08	0.8	8	80	800
$a_1 = 9$	0.09	0.9	9	90	900

[1] A k nim i čísla s opačným znaménkem.

Reprezentace čísel s pohyblivou desetinnou čárkou

Remark 11

Množina $\mathbb{F}(\beta, t, L, U)$ *není uzavřena vůči základním aritmetickým operacím sčítání, odčítání, násobení a dělení.*

Example 12

$x_1 = 1 \in \mathbb{F}(10, 1, -2, 2)$, $x_2 = 0.1 \in \mathbb{F}(10, 1, -2, 2)$, ale $x_1 + x_2 = 1.1 \notin \mathbb{F}(10, 1, -2, 2)$.

Remark 13

Abychom dokázali výsledek zpracovat, musíme provést **zaokrouhlení** *na nejbližší číslo v* $\mathbb{F}(10, 1, -2, 2)$ *tj.* 1*. Tak vznikají* **zaokrouhlovací chyby** *– rounding errors. Takto definované operace, tj. nejprve přesně provedená operace a následné zaokrouhlení do množiny* $\mathbb{F}$ *budeme označovat s indexem fl. Například* $+_{fl}$, $-_{fl}$ *apod.*

Zaokrouhlovací chyby

Example 14

Mějme $\mathbb{F}(10, 2, -5, 5)$ a počítejme součet

$$10 + \underbrace{0.1 + 0.1 \ldots + 0.1}_{10\times}.$$

Postupně dostáváme

$$10 +_{fl} 0.1 = 10$$

a celý výsledek je nakonec 10.

Zaokrouhlovací chyby

Example 15

Mějme $\mathbb{F}(10, 2, -5, 5)$ a počítejme součet

$$\underbrace{0.1 + \ldots + 0.1}_{10\times} + 10.$$

Postupně dostáváme

$$0.1 +_{fl} 0.1 = 0.2$$

a na konec

$$1.0 + 10 = 11.$$

Zaokrouhlovací chyby

Remark 16

Z příkladu vidíme, že pokud sčítáme řádově hodně rozdílná čísla, mantisa nedokáže tento rozdíl pokrýt a větší číslo "přebije" menší.

Zaokrouhlovací chyby

- jednoduchá přesnost bývá pro inženýrské úlohy často málo přesná
- dvojitá přesnost se ukazuje pro většinou reálných úloh dostatečná
- vycházíme z toho, že v reálných úlohách pracujeme s čísly přibližně stejných řádů
  - tj. "buď na úrovni atomů nebo galaxií"

Stabilita úlohy

- viděli jsme, že během výpočtu se nevyhneme vzniku chyb a nepřesností
- stejně tak vstupní data jsou často naměřená experimentálně a jsou poškozenou chybou měření
- bude nás zajímat, jak se to může projevit na konečném řešení

Stabilita úlohy

Předpokládáme, že řešíme úlohu

$$F\left(\vec{x}, \vec{d}\right) = \vec{0}$$

- F je funkční zobrazení vyjadřující úlohu
- $\vec{d}$ jsou data nebo parametry úlohy
- $\vec{x}$ je řešení úlohy

Stabilita úlohy

Example 17

- řešení lineárního systému - $\mathbb{A}\vec{x} = \vec{b} \Leftrightarrow \mathbb{A}\vec{x} - \vec{b} = \vec{0}$ a tedy $F\left(\vec{x}, \vec{d}\right) = \mathbb{A}\vec{x} - \vec{b} = \vec{0}$, kde $\vec{d} \equiv \left\{\mathbb{A}, \vec{b}\right\}$
- výpočet inverzní matice - $\mathbb{A}\mathbb{A}^{-1} = \mathbb{I} \Leftrightarrow \mathbb{A}\mathbb{A}^{-1} - \mathbb{I} = \theta$ a tedy $F\left(\vec{x}, \vec{d}\right) = \mathbb{A}\mathbb{A}^{-1} - \mathbb{I} = \vec{0}$, kde $\vec{d} = \mathbb{A}$ a $\vec{x} = \mathbb{A}^{-1}$
- výpočet vlastních čísel - hledáme $\lambda_1, \ldots, \lambda_n$, tj.

$$\begin{aligned} \det\left(\mathbb{A} - \lambda_1 \mathbb{I}\right) &= 0, \\ &\vdots \quad , \\ \det\left(\mathbb{A} - \lambda_n \mathbb{I}\right) &= 0, \end{aligned}$$

tj. $\vec{x} \equiv \{\lambda_1, \ldots, \lambda_n\}$ a $\vec{d} \equiv \mathbb{A}$.

Stabilita úlohy

Definition 18

Řekneme, že úloha je **dobře položená** (well posed) nebo **stabilní** (stable), pokud má jednoznačné řešení $\vec{x}$, které závisí spojitě na vstupních datech $\vec{d}$ a pro jejich malou změnu $\delta\vec{d}$ existuje řešení

$$F(\vec{x} + \delta\vec{x}, \vec{d} + \delta\vec{d}) = 0,$$

kde $\delta\vec{x}$ je malé. Tj. **malá změna vstupních dat způsobí malou změnu řešení**.
Pokud úloha nesplňuje výše uvedené, pak říkáme, že je **špatně položená** (ill posed) nebo **nestabilní** (unstable).

Stabilita úlohy

Definition 19

Pro regulární čtvercovou matici $\mathbb{A} \in \mathbb{C}^{n,n}$, definujeme **číslo podmíněnosti matice** jako

$$\kappa(\mathbb{A}) = \|\mathbb{A}\| \left\|\mathbb{A}^{-1}\right\|.$$

Je-li $\kappa(\mathbb{A})$ malé, řekneme, že matice je dobře podmíněná. Jinak je špatně podmíněná.

Podmíněnost matic

Example 20

Mějme soustavu

$$20x + 16y + 9z = 45$$
$$14x + 15y + 11z = 40$$
$$13x + 17y + 14x = 44$$

Řešením ve vektor $(1, 1, 1)^T$.

- Pokud pravou stranu změníme o 0.1 na $\vec{b} = (45.1, 39.9, 44.1)^T$, řešením je $\vec{x} = (-12.5, 32, -24.1)^T$.
- Pokud pravou stranu změníme o 0.01 na $\vec{b} = (45.01, 39.99, 44.01)^T$, řešením je $\vec{x} = (-0.25, 4.1, -1.51)^T$.

Podmíněnost matic

Theorem 21

Nechť $\mathbb{A} \in \mathbb{C}^{n,n}$ je regulární. Buď $\vec{x}$ řešení soustavy $\mathbb{A}\vec{x} = \vec{b} \neq \vec{0}$ a perturbace $\delta\vec{x}, \delta\vec{b}$ takové, že platí $\mathbb{A}\left(\vec{x} + \delta\vec{x}\right) = \vec{b} + \delta\vec{b}$. Pak platí

$$\frac{\left\|\delta\vec{x}\right\|}{\left\|\vec{x}\right\|} \leq \kappa\left(\mathbb{A}\right) \frac{\left\|\delta\vec{b}\right\|}{\left\|\vec{b}\right\|},$$

a jde-li o indukovanou maticovou normu, pak existuje nenulový vektor $\vec{b}$ a nenulová perturbace $\delta\vec{b}$ takové, že nastává rovnost.

Důkaz.

Video na Youtube

Numerické metody

Podobně jako pro úlohy, definujeme stabilitu i pro numerické metody.

Definition 22

Řekneme, že numerická metoda je stabilní, pokud při její aplikaci na **stabilní úlohu** způsobí malá změna vstupních parametrů jen malou změnu výsledku.

Remark 23

Malá změna vstupních parametrů je často způsobena zaokrouhlovacími chybami počítačové aritmetiky.

Numerické metody

Základní dělení numerických metod je na:

- přímé
- iterační

Numerické metody - přímé

- po konečném počtu kroků dávají teoreticky přesné řešení
- prakticky ale díky konečné přesnosti počítačové aritmetiky a zaokrouhlovacím chybám dávají pouze přibližné řešení
- řešení získáme až v posledním kroku, do té doby nemáme nic
- jsou robustnější = fungují pro širší třídu úloh
- lze u nich lépe předpovědět zda budou fungovat nebo ne
- z těchto důvodu jsou upřednostňovány v některých kritických průmyslových aplikacích

Numerické metody - iterační

- k přesnému řešení pouze konvergují a to jen teoreticky
- výsledkem metody je posloupnost $\left\{\vec{x}^{(k)}\right\}_{k=1}^{\infty}$ zpřesňujících se aproximací přesného řešení
- jedna iterace by měla být napočítaná mnohem rychleji, než výpočet přímé metody řešící stejný problém
- iterační metody jsou většinou snazší na implementaci
- máme-li dobrý počáteční odhad přesného řešení, může být iterační metoda výrazně rychlejší
- většina numerických metod je iteračních

Analýza iteračních numerických metod

- pro iterační metody je velmi důležité umět odhadnout chybu aproximace řešení jinak ani nevíme, kdy ukončit výpočet
- využijeme vztahu

$$\vec{x}^{(k)} \to \vec{x} \Rightarrow F\left(\vec{x}^{(k)}, \vec{d}\right) \to \vec{0}$$

k následující definici

Definition 24

Buď $\|\cdot\|$ nějaká vektorová norma, pak číslo

$$r^{(k)} = \left\| F\left(\vec{x}^{(k)}, \vec{d}\right) \right\|,$$

nazýváme **reziduem k-té aproximace**.

Analýza iteračních numerických metod

Remark 25

- *pro řešení soustavy* $\mathbb{A}\vec{x} = \vec{b}$ *máme*

$$r^{(k)} = \left\| F\left(\vec{x}^{(k)}, \vec{d}\right) \right\| = \left\| \mathbb{A}\vec{x} - \vec{b} \right\|.$$

- *bohužel neplatí, že je-li malé* $r^{(k)}$*, je malé i* $\left\| \vec{x}^{(k)} - \vec{x} \right\|$
- *reziduum je ale většinou to jediné, co máme*

Analýza iteračních numerických metod

- pokud provádíme analýzu pouze na základě znalosti F a $\vec{d}$, jde o **apriorní odhad**, tj. takový, který lze provést před výpočtem vlastní metody
- pokud provádíme analýzu i na základě znalosti $\vec{x}^{(k)}$, jde o **aposteriorní odhad**, tj. takový, který lze provést až po výpočtu několika iterací vlastní metody

Apriorní odhady

- apriorní odhady mají spíše teoretický význam
- obecně takový odhad vypadá takto

$$\left\| \vec{x}^{(k)} - \vec{x} \right\| \leq C\left(F, \vec{d}\right) \left\| \vec{x}^{(k-1)} - \vec{x} \right\|^{r},$$

kde, $C\left(F, \vec{d}\right) > 0$ a $r \in N^{+}$ je řád metody.

Apriorní odhady

- r udává řád metody
- např. u metody druhého řádu stačí napočítat polovinu iterací k získání stejně přesné aproximace v porovnání s metodou prvního řádu
- metody vyšších řádů než dva bývají v praxi často velmi nestabilní a např. špatný počáteční odhad $\vec{x}^{(0)}$ může způsobit, že metoda ani nekonverguje
- obecně panuje snaha vytvářet metody, které dávají dobrý odhad po pár iteracích
- **v praxi je ale tím nejdůležitějším měřítkem doba výpočtu nutná pro dosažení požadované přesnosti !!!**
  - lze sestrojit metody, které konvergují po pár krocích, ale jedna iterace trvá celou věčnost

Aposteriorní odhady

- aposteriorní odhady využívají navíc znalosti $\vec{x}^{(k)}$, tj. mají tvar

$$\left\| \vec{x}^{(k)} - \vec{x} \right\| \leq C \left(F, \vec{d}, \vec{x}^{(k-1)} \right) \left\| \vec{x}^{(k-1)} - \vec{x} \right\|^r ,$$

- ze své podstaty by měly být přesnější než apriorní odhady
- mají dobré praktické použití
  - bývají základem adaptivních metod, které se umějí přizpůsobit konkrétní řešené úloze

Zdroje chyb počítačových simulací

Chyby, které vznikají při řešení reálných úloh...

Reálný problém $\xrightarrow[\text{Chyba modelu}]{\text{Matemat. popis}}$ matematický model $\xrightarrow[\text{Chyba diskretizace}]{\text{Diskretizace}}$ numerický model (algebraický model) $\xrightarrow[\text{Chyba linearizace}]{\text{Linearizace}}$ lineární algebraický model $\xrightarrow[\text{Zaokrouhlovaci chyby}]{\text{Reseni}}$ aproximace řešení.

... lze rozdělit do čtyř skupin:

Zdroje chyb počítačových simulací

- chyby způsobené samotným modelem - fyzikálním, matematickým
- chyby vzniklé při měření vstupních dat pomocí experimentů
- chyby vzniklé diskretizací tj. náhradou nekonečných limit konečným počtem operací
- chyby vzniklé zaokrouhlováním z důvodů aritmetiky s konečnou přesností

Otázky

- reprezentace čísel s pohyblivou desetinnou čárkou
- jednoduchá a dvojitá přesnost
- **zaokrouhlovací chyby**
- **podmíněnost matice**
- **řád metody**