# Přednáška 4 – aritmetika PC

→ opáčko z minule:

- pro Chebyshev. interpolaci jsme dostali přesnost pouze na úrovni cca $10^{-16}$
- pro Vandermond. interpolaci a Lagrange interpolaci jsme dostali dost jiné výsledky v PC, přestože jsou matem. ekvivalentní
- pro barycentrického Lagrange jsme dostali lepší výsledky než pro klasického, přestože jsou matem. ekvivalentní (barycentrický Lagr. – avika)

---

## Jak jsou čísla uložená v PC?

→ existuje standardizace (od 1985 – do té doby divoký západ, nekompatibilita a soutěž trhu o lepší design)

IEEE 754 (wiki)

→ spousta 0 a 1, ale nutně konečně mnoho

⇒ nerealistické uložit přesně $\pi$, $\sqrt{2}$, $e$, … ale dokonce nezvládáme ani lehčí

→ nepůjdeme do detailů, pouze ukážu tři konkrétní čísla

$x \in \mathbb{R}$ .... číslo co chci reprezentovat v PC

$x_{FPA} \in \mathbb{R}$ .... to číslo, které opravdu v PC mám

(FPA ≡ floating-point precision arithmetic)

někdy finite precision arithmetic

⇝ předpokládáme, že $x = x_{FPA}$ nebo $x \approx x_{FPA}$

$x = 6 \cdot 10^{+1} + 1 \cdot 10^{0} = 61$ .... dekadický zápis

$x = 32 + 16 + 8 + 4 + 1 = 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$

$x_{FPA} = 111101$ .... binární zápis

⇝ a $\{0,1\}$ my v PC umíme

⇝ opravdu $x = x_{FPA}$

---

$$x = 0.15625$$
$$= (+1) \cdot 10^{-1} \cdot 1.5625 = (+1) \cdot 10^{-1} \cdot \left(1 \cdot 10^{0} + 5 \cdot 10^{-1} + 6 \cdot 10^{-2} + 2 \cdot 10^{-3} + 5 \cdot 10^{-4}\right)$$
$$= (+1) \cdot 2^{-3} \cdot 1.25 = (+1) \cdot 2^{-3} \cdot \left(1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}\right)$$
$$= (+1) \cdot 2^{exp} \cdot [1\,0\,1] \quad \& \quad exp = -[1\,1]'' = -(1 \cdot 2^1 + 1 \cdot 2^0)$$
$$= (+1) \cdot \text{"}-[1\,1]\text{"} \cdot \text{"}[1\,0\,1]\text{"}$$

$x_{FPA} = \{1\}, -\{1,1\}, \{1\,0\,1\}$ ⟹ $x = x_{FPA}$

---

$$x = 1.3 = (+1) \cdot 10^{0} \cdot 1.3 = (+1) \cdot 10^{0} \cdot \left(1 \cdot 10^{0} + 3 \cdot 10^{-1}\right)$$
$$= (+1) \cdot 2^{0} \cdot \left(1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 0 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6} + 0 \cdot 2^{-7} + 0 \cdot 2^{-8} + 1 \cdot 2^{-9} + 1 \cdot 2^{-10} + \dots\right)$$
$$= (+1) \cdot 2^{0} \cdot [1\,0\,1\,00\,11\,00\,11\,00\,11 \dots]$$

⟹ $x_{FPA} \neq x$ .... musíme ten nekonečný binární rozvoj někde zaříznout

Standard ve všech PC: tzv. double precision ≡ fp64 formát čísel

≡ {1 bit znaménko, 11 bitů exponent, 52 bitů na binární rozvoj}

<u>Platí</u>: $\forall x \in (-10^{308}, -10^{-38}) \cup (10^{-38}, 10^{308})\ \exists\, x_{FPA} : \frac{|x - x_{FPA}|}{|x|} \leq 2 \cdot 10^{-16}$

Všechny operace v PC se provádějí pouze s $x_{FPA}$, tj. rovnou na těch lístcích $\{0,1\}^{64}$.

> všechna data & naše manipulace/výpočty s nimi jsou (snad jen trochu) nepřesné/špatné

## Potenciální problém 1

- existují matematické problémy pro které malá perturbace (= změna) vstupních dat/parametrů může vést k velké změně v přesném řešení toho problému –> –> tzv. efekt motýlých křídel
- takovým problémům se říká „špatně podmíněné" (= ill-conditioned) opakem jsou „dobře podmíněné" problémy

=> podmíněnost je vlastnost problému a pro extrémně špatně podmíněné problémy je jen malá naděje na numerický výpočet řešení.

## Potenciální problém 2

- tu nepříjemnou vlastnost „efektu motýlých křídel" může mít ale také námi navržený numerický algoritmus: zaokrouhlovací chyby se mohou akumulovat a i pro dobře podmíněný problém dát zkreslenou/špatnou odpověď.
- obecně mluvíme o stabilitě algoritmů, viz níže.

## Obecná strategie pro analyzování:

**Krok 1:** analyzujeme matematický problém a zjistíme jeho podmíněnost = citlivost řešení na změnu vstupních dat/parametrů.

**Krok 2:** analyzujeme náš algoritmus jako „přesný výpočet pro nepřesná data"

Příklad – algoritmus pro sčítání 2 skalárů $\alpha, \beta \in \mathbb{R}$:

tužka a papír: $\alpha + \beta = \gamma$ vs. algoritmus: $\alpha +_{PC} \beta = \hat{\gamma} \approx \gamma$

analýza stability: dokážeme existenci $\tilde{\alpha}, \tilde{\beta} \in \mathbb{R}$ t., že

- $\tilde{\alpha} + \tilde{\beta} = \hat{\gamma}$
- $\tilde{\alpha} \approx \alpha$
- $\tilde{\beta} \approx \beta$

Pak algoritmus je „stabilní" (= zpětně stabilní), pokud

$$\frac{\max\{|\alpha - \tilde{\alpha}|, |\beta - \tilde{\beta}|\}}{\max\{|\alpha|, |\beta|\}} \approx C \cdot \varepsilon_{mach}$$

**Krok 3:** $x_{PC}$ = „výstup ze stabilního algoritmu pro dobře podmíněný problém"

stabilní algoritmus //

$x_{PC}$ = „výstup z výpočtu „tužka+papír" pro „trochu změněná" vstupní data" $\xRightarrow{\text{dobře podmíněný problém}}$ $x_{PC} \approx x_{\text{tužka-a-papír}}$