Numerická lineárna algebra. Zobrazenie reálnych čísiel v počítači

Ing. Gabriel Okša, CSc.

Matematický ústav
Slovenská akadémia vied
Bratislava

Stavebná fakulta STU

Obsah

Predmet NLA

- Numerická lineárna algebra sa zaoberá návrhom algoritmov a analýzou ich vlastností pre problémy spojitej (klasickej) matematiky, pričom sa používajú nástroje lineárnej algebry (L. Trefethen, Oxford, 1997).
- Algoritmus: Návod, ako zo vstupných údajov (množiny čísiel) dospieť k výstupným údajom (iná množina čísiel). Algoritmus môže byť konečný (napr. QR rozklad matice) alebo nekonečný (napr. výpočet vlastných čísiel matice rádu n pre $n > 5$).
- Analýza vlastností: presnosť, stabilita, robustnosť, ...
- Spojitá matematika: Problém je formulovaný pomocou reálnych alebo komplexných premenných. Opakom je diskrétna matematika (napr. teória grafov), kde vystupujú celočíselné premenné.

Floating-Point Number System - 1/4

- Každý počítač má ohraničenú pamäť $\Rightarrow$ niektoré reálne čísla nemôžu byť reprezentované v počítači presne.
- IEEE Floating-Point Standard (1985): číslicový systém s pohyblivou rádovou čiarkou je definovaný pomocou usporiadnej štvorice celých čísiel (β, t, L, U), kde:
  β je základ (báza, radix), t je presnosť, L je dolná hranica exponentu, U je horná hranica exponentu.
- Nenulové normalizované číslo v systéme (β, t, L, U) má tvar:

  $$\pm .d_1 d_2 \cdots d_t \times \beta^e = \pm(d_1 \beta^{-1} + d_2 \beta^{-2} + \cdots + d_t \beta^{-t}) \beta^e,$$

  kde $\pm$ je znamienko,
  $.d_1 d_2 d_3 \cdots d_t$ je mantisa ($\mathbf{1} \leq d_1 \leq \beta - 1$, $\mathbf{0} \leq d_i \leq \beta - 1$ pre $2 \leq i \leq t$),
  e je exponent ($L \leq e \leq U$).

Floating-Point Number System - 2/4

- **Príklad**: $\beta = 2$, $L = -126$, $U = 127$, $t = 24$ (aj so znamienkom) ... jednoduchá presnosť (IEEE);
  $\beta = 2$, $L = -1022$, $U = 1023$, $t = 53$ (aj so znamienkom) ... dvojitá presnosť (IEEE).
- IEEE FPS požaduje, aby exponent mal k dispozícii ešte dve skryté hodnoty:
  $L - 1$: používa sa na kódovanie ± 0 a denormalizovaných čísiel s $d_1 \neq 1$ (pri $\beta = 2$).
  $U + 1$: používa sa na kódovanie výsledku, ktorý sa v číselnom systéme počítača nedá zobraziť (tzv. NaN = 'Not a Number', tiež sa označuje ako $\pm\infty$).
- Normalizované binárne čísla ($\beta = 2$): v počítači sa ukladá mantisa (t bitov), znamienko a exponent s posunom tak, aby exponent bol vždy nezáporný (t.j. pričíta sa $|L| + 1$).

Floating-Point Number System - 3/4

- **Príklad**: jednoduchá presnosť: číslo je reprezentované 32 bitmi (1 bit na znamienko, 8 bitov na exponent, 23 bitov na mantisu);
  dvojitá presnosť: potrebujeme 64 bitov (1 bit na znamienko, 11 bitov na exponent, 52 bitov na mantisu).
- Označme F_t množinu normalizovaných čísiel s pohyblivou rádovou čiarkou s presnosťou t. Potom F_t nie je uzavretá vzhľadom na základné aritmetické operácie: sčítanie, odčítanie, násobenie a delenie.
- **Príklad**: $\beta = 10$, $t = 3$, $L = -1$, $U = 2$; dané $a = 0.112 \times 10^2$, $b = 0.113 \times 10^1$. Potom $c = a \times b = 0.12656 \times 10^2$ nie je v F_t (mantisa obsahuje viac ako $t = 3$ číslice).

Floating-Point Number System - 4/4

- Vypočítané číslo nemusí byť v F_t, pretože:
  1. Exponent padne mimo interval $[L, U] \Rightarrow$ podtečenie (underflow) resp. pretečenie (overflow).
     Pretečenie: toto je obvykle vážny problém pre väčšinu počítačov. Výsledkom pretečenia je NaN ('Not a Number'). Niektoré programovacie jazyky (napr. C++) umožňujú programátorovi "odchytiť" pretečenie a definovať ďalší postup.
     Podtečenie: menej vážne ako pretečenie. Výsledkom je buď nula, denormalizované číslo alebo $\pm 2^L$.
  2. Mantisa výsledku obsahuje viac ako t číslic $\Rightarrow$ nutná je nejaká forma zaokrúhlenia výsledku - t.j. výsledok sa následne upraví do normalizovaného tvaru.
- Pretečeniu a podtečeniu sa niekedy dá zabrániť reorganizáciou výpočtu. Napr. $\|x\| = \sqrt{x_1^2 + \cdots + x_n^2}$:
  1. $m = \max(|x_1|, \cdots, |x_n|)$, 2. $y_i = x_i/m$,
  3. $\|x\| = m\sqrt{y_1^2 + \cdots + y_n^2}$.

Zaokrúhľovacie chyby - 1/3

- Ak mantisa výsledku aritmetickej operácie obsahuje viac ako t číslic, potom sa dá upraviť do tvaru reprezentovateľného v počítači dvomi spôsobmi:
  1. Odtrhnutie ('chopping'): číslice za d_t sa jednoducho "zahodia".
  2. Zaokrúhlenie ('rounding'): číslica d_t sa zaokrúhli nahor (ak $d_{t+1} \geq \beta/2$) alebo nadol (ak $d_{t+1} < \beta/2$), a číslice za d_t sa "zahodia".
- Oba postupy aproximujú vypočítané číslo. Aproximácia nesie so sebou vždy chybu. Nech $\hat{x}$ je aproximácia čísla x. Potom:

  $|\hat{x} - x|$ je absolútna chyba aproximácie;

  $\frac{|\hat{x}-x|}{|x|}$ je relatívna chyba aproximácie.

  Relatívna chyba berie do úvahy veľkosť aproximovaného čísla x a dáva informáciu o počte signifikantných číslic v aproximácii.

Zaokrúhľovacie chyby - 2/3

- **Definícia**: Hovoríme, že $\hat{x}$ aproximuje x na s signifikantných číslic, ak je s je najväčšie nezáporné číslo, pre ktoré je relatívna chyba aproximácie $|\hat{x} - x|/|x| < 5 \times 10^{-s}$.
- **Veta 1**: Nech $\mathrm{fl}(x)$ označuje reprezentáciu čísla x v pohyblivej rádovej čiarke. Potom:

$$\frac{|\mathrm{fl}(x) - x|}{|x|} \leq \mu, \tag{1}$$

kde $\mu = 0.5 \times \beta^{1-t}$ pre 'rounding', resp. $\mu = \beta^{1-t}$ pre 'chopping'.

- **Definícia**: Číslo μ sa nazýva jednotka zaokrúhľovania ('unit roundoff') a $\mu_M = 2\mu$ je presnosť stroja ('machine precission').

Zaokrúhľovacie chyby - 3/3

- Presnosť stroja μ_M je najmenšie kladné číslo v pohyblivej rádovej čiarke také, že $\mathrm{fl}(1+\mu_M) > 1$.
- Veta 1 hovorí, že pre dostatočne veľké t je množina F_t ('floating-point numbers') dostatočne "hustá". Napr. pre dvojnásobnú presnosť s $t = 53$ je $\mu = 2^{-53} \approx 1.11 \times 10^{-16}$.
- Napriek tejto "hustote" je F_t vždy diskrétna množina. Napr. pre dvojnásobnú presnosť s $t = 53$:

  Interval $[1,2] \Rightarrow$ v F_t sú čísla:
  $1,\ 1+2^{-52},\ 1+2\times 2^{-52},\ 1+3\times 2^{-52},\ \ldots,\ 2.$ (medzera $2^{1-t} = 2^{-52}$)
  Interval $[2,4] \Rightarrow$ v F_t sú čísla:
  $2,\ 2+2^{-51},\ 2+2\times 2^{-51},\ 2+3\times 2^{-51},\ \ldots,\ 4.$ (medzera $2^{2-t} = 2^{-51}$)
  Interval $[2^j, 2^{j+1}]$: zober $[1,2]$ a vynásob 2^j. (medzera 2^{j+1-t})

Štruktúra množiny F_t

- Takže medzi susednými číslami v F_t absolútna medzera rastie s mocninou 2, ale relatívna medzera nie je nikdy väčšia ako $2^{-52} = 2\mu = \mu_M \approx 2.22 \times 10^{-16}$.
- **Rozsah** $y \in F_t$ (normalizované): $\beta^{L-1} \leq |y| \leq \beta^U(1 - \beta^{-t})$.
- Najmenšie normalizované kladné číslo v F_t je $\lambda \equiv \beta^{L-1}$. V intervale $(0, \lambda]$ ležia denormalizované čísla s $d_1 = 0$. Najmenšie kladné denormalizované číslo je $\tau \equiv \beta^{L-t} = \lambda\,\mu_M$. Absolútna medzera v intervale $(0, \lambda]$ je $\beta^{L-t} = \lambda\,\mu_M$.
  Denormalizované čísla nepatria do F_t (majú menej ako t signifikantných číslic), ale rozširujú F_t.
- Absolútna medzera v intervale $(\beta^j\lambda, \beta^{j+1}\lambda]$ je β^{L+j-t} pre $j = 0, 1, \ldots, -L - 1$.
- Absolútna medzera v intervale $(\beta^{-1}, 1]$ je $\beta^{-t} = \mu_M/\beta$.

‘Floating-point’ aritmetika - 1/2

- Nerovnosť (1) znamená, že pre každé $x \in \mathbb{R}$ existuje $\hat{x} \in F_t$ také, že $|\hat{x} - x| \leq \mu |x|$.
- **Ešte inak**: Nech $\mathrm{fl} : \mathbb{R} \to F_t$ je zobrazenie, ktoré každému $x \in \mathbb{R}$ pripradí najbližšiu floating-point reprezentáciu pomocou zaokrúhľovania. Potom pre každé $x \in \mathbb{R}$ existuje $\epsilon,\ |\epsilon| \leq \mu$, že
$$\mathrm{fl}(x) = x(1 + \epsilon). \tag{2}$$
- Základný axióm floating-point aritmetiky splňujúcej IEEE Standard: Nech $\{+, -, \times, /\}$ sú základné aritmetické operácie nad reálnymi číslami. Nech označenie fl pred niektorou z týchto operácií znamená, že je to binárna operácia nad dvomi číslami v F_t a výsledok je opäť v F_t. Potom pre každé $x, y \in F_t$ existuje $\epsilon,\ |\epsilon| \leq \mu$ tak, že
$$\mathrm{fl}(x \star y) = (x \star y)(1 + \epsilon), \tag{3}$$
kde $\star$ je jedna z operácií $\{+, -, \times, /\}$.

‘Floating-point’ aritmetika - 2/2

- Nech daný počítač spĺňa IEEE Floating-Point Standard (1985). Potom rovnica (2) hovorí o zobrazení čísla x s relatívnou presnosťou ϵ, zatiaľčo rovnica (3) hovorí o vykonaní ľubovoľnej aritmetickej operácie s relatívnou presnosťou ϵ. V oboch prípadoch je ϵ “malé”, maximálne sa rovná jednotke zaokrúhľovania: $|\epsilon| \leq \mu$.
- Základný axióm floating-point aritmetiky treba chápať ako požiadavku na počítač, ak chce spĺňať IEEE Standard. Týka sa to hardvéru i firmvéru, ktorý má na starosti výpočet aritmetických operácií. Niekedy (a čoraz častejšie) je podobná relatívna presnosť garantovaná aj pre výpočet druhej odmocniny (čo je unárna operácia):

$$x \in F_t, x > 0 \Rightarrow \mathrm{fl}(\sqrt{x}) = \sqrt{x}(1+\epsilon),\ |\epsilon| \leq \mu.$$

Ochranná číslica ('Guard Digit')

- Relatívna chyba 4 aritmetických operácií má byť najviac μ. Pokiaľ sa však sčítanie alebo odčítanie robí bez tzv. ochrannej číslice, potom výsledok nemusí spĺňať uvedenú presnosť.
- **Definícia**: Ochranná číslica je jedna číslica na mieste d_{t+1}, ktorej úlohou je zachytiť najmenšiu mocninu základu pri zarovnaní operandov, ktorá by sa inak stratila.
- **Príklad**: $\beta = 10,\ t = 3,\ \mu = 0.005$. Úloha: $\mathrm{fl}(x + y)$ pre $x = 0.101 \times 10^2,\ y = -0.994 \times 10^1$.
  Výpočet s ochrannou číslicou:
  $\mathrm{fl}(x + y) = (0.101\mathbf{0} - 0.099\mathbf{4}) \times 10^2 = 0.001\mathbf{6} \times 10^2 = 0.160 \times 10^0 = (x + y)(1 + \epsilon_1)$, kde $\epsilon_1 = 0$.
  Výpočet bez ochrannej číslice:
  $\mathrm{fl}(x + y) = (0.101 - 0.099) \times 10^2 = 0.002 \times 10^2 = 0.200 \times 10^0 = (x + y)(1 + \epsilon_2)$, kde $\epsilon_2 = 0.25 = 50\mu$.

Príklady

1. Suma n floating-point čísiel: Je daných n čísiel $x_1,\ x_2\ \ldots, x_n$. Odhadnite chybu súčtu $s_n = \sum_{i=1}^{n} x_i$ v pohyblivej rádovej čiarke.

$s_2 = \mathrm{fl}(x_1 + x_2) = (x_1 + x_2)(1 + \delta_2),\ |\delta_2| \leq \mu \ \Leftrightarrow$

$s_2 - (x_1 + x_2) = (x_1 + x_2)\,\delta_2$;

Krok i: $s_{i+1} = \mathrm{fl}(s_i + x_{i+1}),\ i = 2, 3, \ldots, n-1$;

$s_3 = \mathrm{fl}(s_2 + x_3) = (s_2 + x_3)(1 + \delta_3) = [(x_1 + x_2)(1 + \delta_2) + x_3](1 + \delta_3) = (x_1 + x_2)(1 + \delta_2 + \delta_3 + \delta_2\delta_3) + x_3 + x_3\delta_3 \approx x_1 + x_2 + x_3 + (x_1 + x_2)\,\delta_2 + (x_1 + x_2 + x_3)\,\delta_3 \ \Leftrightarrow$

$s_3 - (x_1 + x_2 + x_3) \approx (x_1 + x_2)\,\delta_2 + (x_1 + x_2 + x_3)\,\delta_3$; atď;

Až nakoniec:

$s_n - (x_1 + x_2 + \cdots + x_n) \approx$

$(x_1 + x_2)\,\delta_2 + (x_1 + x_2 + x_3)\,\delta_3 + \cdots + (x_1 + x_2 + \cdots + x_n)\,\delta_n,$

takže:

$\mathrm{fl}(x_1 + x_2 + \cdots + x_n) - (x_1 + x_2 + \cdots + x_n) \approx x_1(\delta_2 + \delta_3 + \cdots + \delta_n) + x_2(\delta_2 + \delta_3 + \cdots + \delta_n) + x_3(\delta_3 + \cdots + \delta_n) + \cdots + x_n\delta_n, \quad |\delta_i| \leq \mu.$

Príklady

Keď je n čísiel zoradených vzostupne (t.j. $|x_1| \leq |x_2| \leq \cdots \leq |x_n|$), potom s najmenšími číslami je spojená najväčšia chyba $\Rightarrow$ celková chyba súčtu bude najmenšia.

2. Súčin n floating-point čísiel: Nech je daných n reálnych nenulových čísiel $x_1, x_2 \ldots, x_n$. Ukážte, že:
$\mathrm{fl}(\prod_{i=1}^{n} x_i) \approx (1+\epsilon) \prod_{i=1}^{n} x_i$, kde
$1+\epsilon = (1+\delta_2)(1+\delta_3)\cdots(1+\delta_n)$ a $|\delta_i| \leq \mu,\ 2 \leq i \leq n$.

3. Predpokladajme že $(n-1)\,\mu < 0.1$. Ukážte, že v Príklade 2 je $\epsilon < 1.06\,(n-1)\,\mu$ (ϵ je relatívna chyba výpočtu súčinu).

4. Skalárny súčin dvoch vektorov: Nech x a y sú dva n-rozmerné vektory. Ukážte, že:
$\mathrm{fl}(x^T y) = \sum_{i=1}^{n} x_i y_i\,(1+\epsilon_i)$, kde
$1+\epsilon_i = (1+\delta_i)(1+\eta_i)(1+\eta_{i+1})\cdots(1+\eta_n)$ a $|\delta_i|,\ |\eta_i| \leq \mu$.