Alkalmazott matematika

Baran Ágnes

Lebegőpontos számok

Mennyire bízhatunk meg a gépünk által kiszámolt eredményekben?

1. feladat (labor)

Vizsgálja meg számítógépén a $0.4 - 0.5 + 0.1 == 0$ logikai kifejezés értékét! Mi lesz a $0.1 - 0.5 + 0.4 == 0$ logikai kifejezés értéke?

2.feladat (labor)

Az alábbi algoritmus végrehajtása után mennyi az x elméleti, illetve a gépi számítás után adódó értéke?

```
x=1/3;
for i=1:40
   x=4*x-1;
end
```

3. feladat (labor)

Vizsgálja meg számítógépén a $2^{66} + 1 == 2^{66}$, $2^{66} + 10 == 2^{66}$, $2^{66} + 100 == 2^{66}$, $2^{66} + 1000 == 2^{66}$ és $2^{66} + 10000 == 2^{66}$ logikai kifejezések értékét!

4. feladat (labor)

Mit tapasztal, ha az alábbi kódokat lefuttatja?

```
a=0;
for i=1:5
   a=a+0.2;
end
a==1
```

```
a=1;
for i=1:5
   a=a-0.2;
end
a==0
```

Próbáljuk megmagyarázni a tapasztalt jelenségeket!

Lebegőpontos számok

Példa.

$a = 10$

$$0.3721 = \frac{3}{10} + \frac{7}{10^2} + \frac{2}{10^3} + \frac{1}{10^4}$$

$$21.65 = 0.2165 \cdot 10^2 = \left(\frac{2}{10} + \frac{1}{10^2} + \frac{6}{10^3} + \frac{5}{10^4}\right) \cdot 10^2$$

$a = 2$

$$0.1101 = \frac{1}{2} + \frac{1}{2^2} + \frac{0}{2^3} + \frac{1}{2^4}$$

$$0.001011 = 0.1011 \cdot 2^{-2} = \left(\frac{1}{2} + \frac{0}{2^2} + \frac{1}{2^3} + \frac{1}{2^4}\right) \cdot 2^{-2}$$

Lebegőpontos számok

A nemnulla lebegőpontos számok alakja:

$$\pm a^k \left(\frac{m_1}{a} + \frac{m_2}{a^2} + \cdots + \frac{m_t}{a^t} \right)$$

ahol

$a > 1$ egész, a számábrázolás alapja

$t > 1$, egész, a mantissza hossza

$k_- \leq k \leq k_+$ egész, a karakterisztika, ahol $k_- < 0$ és $k_+ > 0$ adott

$1 \leq m_1 \leq a - 1$, egész (a szám normalizált)

$0 \leq m_i \leq a - 1$, egész, ha $i = 2, \ldots, t$

röviden: $[\pm|k|m_1, \ldots, m_t]$
ahol $(m_1, \ldots, m_t)$ a mantissza.

Az a, t, k_-, k_+ értéke egyértelműen leírja az ábrázolható számok halmazát.

Példa. (gyakorlat)

Legyen $a = 2,\ t = 4,\ k_- = -3,\ k_+ = 2.$

(a) Írjuk fel a következő számok lebegőpontos alakját.

$$0.3125, \quad 0.4375, \quad 1.75, \quad 6.5$$

(b) Hány pozitív normalizált lebegőpontos szám ábrázolható ilyen jellemzők mellett?

Adott a, t, k_-, k_+ esetén
a legnagyobb ábrázolható szám:

$$M_\infty = a^{k_+}\left(\frac{a-1}{a} + \frac{a-1}{a^2} + \cdots + \frac{a-1}{a^t}\right)$$

$$= a^{k_+}\left(1 - \frac{1}{a} + \frac{1}{a} - \frac{1}{a^2} + \cdots + \frac{1}{a^{t-1}} - \frac{1}{a^t}\right)$$

$$= a^{k_+}\left(1 - a^{-t}\right)$$

a legkisebb pozitív normalizált ábrázolható szám:

$$\varepsilon_0 = a^{k_-}\left(\frac{1}{a} + 0 + \cdots + 0\right) = a^{k_- - 1}$$

Szubnormális számok: ha $k = k_-$, akkor $m_1 = 0$ is lehet.

Az 1 mindig lebegőpontos szám:

$$1 = a^1 \cdot \frac{1}{a}, \qquad \text{vagy röviden: } 1 = [+|1|1, 0, \dots, 0]$$

Az 1 jobboldali szomszédja:

$$1 + \varepsilon_1 = [+|1|1, 0, \dots, 0, 1]$$

másképp:

$$1 + \varepsilon_1 = a\left(\frac{1}{a} + 0 + \cdots + 0 + \frac{1}{a^t}\right) = 1 + a^{1-t}$$

azaz $\varepsilon_1 = a^{1-t}$ (**gépi epszilon**)

5. feladat (labor)

(a) Írjon egy kódot a gépi epszilon meghatározására.

(b) Olvassa el az Octave/Matlab eps függvényének help-jét. Nézze meg az eps (azaz az eps(1)) értékét.

6. feladat (labor)

(a) Írjon egy kódot az ε_0 meghatározására.

(b) Nézze meg az `eps(0)` értékét!

7. feladat (labor)

Írassa ki gépén a `realmin` és `realmax` értékét. Vizsgálja meg a `realmin('single')` és `realmax('single')` értékeket is.

Az IEEE lebegőpontos aritmetikai szabvány:

	egyszeres pontosság	dupla pontosság
méret	32 bit	64 bit
mantissza	23+1 bit	52+1 bit
karakterisztika	8 bit	11 bit
ε_1	$\approx 1.19 \cdot 10^{-7}$	$\approx 2.22 \cdot 10^{-16}$
M_∞	$\approx 10^{38}$	$\approx 10^{308}$

mivel m_1 mindig 1, ezért nem ábrázoljuk
az előjel ábrázolására 1 bit

Adott a, t, k_+, k_- mellett az ábrázolható lebegőpontos számok a $[-M_\infty, M_\infty]$ intervallum egy megszámlálható részhalmazát alkotják.

8. feladat (gyakorlat)

(a) Ábrázoljuk számegyenesen az $a = 2$, $t = 4$, $k_- = -3$, $k_+ = 2$ jellemzők mellett felírható összes pozitív normalizált lebegőpontos számot.

(b) A fenti számábrázolási jellemzők mellett mennyi lesz M_∞, ε_0 és ε_1 értéke?

(c) Mit mondhatunk két szomszédos szám távolságáról?

(d) Mit mondhatunk a szomszédos számok távolságáról, ha k_+ értékét 4-re módosítjuk?

(e) Mi lenne, ha $k_+ > 4$ teljesülne?

Példa.

A pozitív normalizált lebegőpontos számok $a = 2$, $t = 4$, $k_- = -3$, $k_+ = 2$ esetén.

	$k=0$	$k=1$	$k=2$	$k=-1$	$k=-2$	$k=-3$
0.1000	$\frac{8}{16}$	$\frac{8}{8}$	$\frac{8}{4}$	$\frac{8}{32}$	$\frac{8}{64}$	$\frac{8}{128}$
0.1001	$\frac{9}{16}$	$\frac{9}{8}$	$\frac{9}{4}$	$\frac{9}{32}$	$\frac{9}{64}$	$\frac{9}{128}$
0.1010	$\frac{10}{16}$	$\frac{10}{8}$	$\frac{10}{4}$	$\frac{10}{32}$	$\frac{10}{64}$	$\frac{10}{128}$
0.1011	$\frac{11}{16}$	$\frac{11}{8}$	$\frac{11}{4}$	$\frac{11}{32}$	$\frac{11}{64}$	$\frac{11}{128}$
0.1100	$\frac{12}{16}$	$\frac{12}{8}$	$\frac{12}{4}$	$\frac{12}{32}$	$\frac{12}{64}$	$\frac{12}{128}$
0.1101	$\frac{13}{16}$	$\frac{13}{8}$	$\frac{13}{4}$	$\frac{13}{32}$	$\frac{13}{64}$	$\frac{13}{128}$
0.1110	$\frac{14}{16}$	$\frac{14}{8}$	$\frac{14}{4}$	$\frac{14}{32}$	$\frac{14}{64}$	$\frac{14}{128}$
0.1111	$\frac{15}{16}$	$\frac{15}{8}$	$\frac{15}{4}$	$\frac{15}{32}$	$\frac{15}{64}$	$\frac{15}{128}$

$M_\infty = 2^2(1 - 2^{-4}) = \frac{15}{4}$ és $\varepsilon_0 = 2^{-3-1} = \frac{1}{16}$ $\left(= \frac{8}{128}\right)$

Legyen $y = a^k \cdot 0.m_1 m_2 ... m_t$.

A legközelebbi nála nagyobb szám

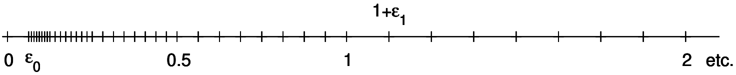
$$a^k \cdot \frac{1}{a^t} = a^{k-t}$$

távolságra van tőle.

Nagyobb karakterisztika $\rightarrow$ nagyobb lépésköz.

Ha $k > t$, akkor a lépésköz nagyobb mint 1.

$a = 2,\ t = 4,\ k_- = -3$ esetén



$\varepsilon_0 = a^{k_- - 1} = 2^{-4} = \frac{1}{16}$,

$\varepsilon_1 = a^{1-t} = 2^{-3} = \frac{1}{8}$

9. feladat (labor)

Vizsgálja meg számítógépén a $2^{66} + 1 == 2^{66}$, $2^{66} + 10 == 2^{66}$, $2^{66} + 100 == 2^{66}$, $2^{66} + 1000 == 2^{66}$ és $2^{66} + 10000 == 2^{66}$ logikai kifejezések értékét! Keresse meg azt a legkisebb $n > 0$ számot, melyre a $2^{66} + n == 2^{66}$ logikai kifejezés értéke hamis. Mennyi az `eps(2^66)` értéke?

Dupla pontosság esetén ($t = 53$):

y	a jobboldali szomszéd távolsága
1	$\approx 2.22 \cdot 10^{-16}$
16	$\approx 3.5527 \cdot 10^{-15}$
1024	$\approx 2.27 \cdot 10^{-13}$
$2^{20} \approx 10^6$	$\approx 2.33 \cdot 10^{-10}$
$2^{52} \approx 4.5 \cdot 10^{15}$	1
$2^{60} \approx 1.15 \cdot 10^{18}$	256
$2^{66} \approx 7.38 \cdot 10^{19}$	16384

Kerekítés

A $[-M_\infty, M_\infty]$ intervallumból nem minden szám írható fel lebegőpontos alakban.

Példa

A 0.1 kettes számrendszerbeli alakja:

$$0.0001100110011001100....$$

Az $\frac{1}{3}$ kettes számrendszerbeli alakja:

$$0.0101010101010....$$

Kerekítés

Legyen $x \in [-M_\infty, M_\infty]$ egy valós szám, $fl(x)$ pedig a hozzárendelt lebegőpontos szám.

Szabályos kerekítés esetén:

$$fl(x) = \begin{cases} 0, & \text{ha } |x| < \varepsilon_0 \\ \text{az } x\text{-hez legközelebbi lebegőpontos számok} & \text{ha } |x| \geq \varepsilon_0 \\ \text{közül a nagyobb abszolút értékű,} & \end{cases}$$

Levágás esetén:

$$fl(x) = \begin{cases} 0, \quad \text{ha } |x| < \varepsilon_0 \\ \text{az } x\text{-hez legközelebbi lebegőpontos szám a 0 felé, ha } |x| \geq \varepsilon_0 \end{cases}$$

Megjegyzés

Ha az ábrázolni kívánt szám két szomszédos lebegőpontos szám között félúton helyezkedik el, akkor a valóságban az előzőnél bonyolultabb kerekítési szabály alapján történik a kerekítés.

Példa

Legyen $a = 2$, $t = 4$, $k_- = -3$, $k_+ = 2$. Mi lesz a 0.1-hez rendelt lebegőpontos szám szabályos kerekítés, illetve levágás esetén?

A 0.1 kettes számrendszerben, normalizálva:

$$2^{-3} \cdot 0.1100110011001100....$$

Szabályos kerekítés:

$$fl(0.1) = 2^{-3} \cdot 0.1101$$

Levágás:

$$fl(0.1) = 2^{-3} \cdot 0.1100$$

10. feladat. (gyakorlat)

Legyen $a = 2$, $t = 4$, $k_- = -3$, $k_+ = 2$. Mi lesz az alábbi számokhoz rendelt lebegőpontos szám szabályos kerekítés, illetve levágás esetén?

$$0.2, \quad 0.6, \quad \frac{1}{32}$$

11. feladat (labor)

Az alábbi algoritmus elméletileg minden $x \geq 0$ esetén az x eredeti értékét adja vissza. Vizsgálja meg mi történik a gyakorlatban, ha az algoritmust $x = 1000$, $x = 100$ kezdőértékkel futtatja! Mi az oka a tapasztalt jelenségnek?

```
for i=1:60
   x=sqrt(x);
end
for i=1:60
   x=x^2;
end
```

12. feladat (labor)

Ismert, hogy $\lim_{x\to 0} \frac{e^x-1}{x} = 1$. Számítsa ki az $\frac{e^x-1}{x}$ hányados értékét egyre csökkenő x értékek esetén! ($x = 1$ kezdőértékkel x-et 40-szer, 200-szor, 2000-szer felezgetve írassa ki a kifejezés értékét!) Magyarázza meg a tapasztalt jelenséget!

13. feladat (labor)

Tekintsük az alábbi azonosságot (ahol $x \neq 0$)!

$$\left(\frac{\frac{1}{x^2}}{10} + 1 \right) x^2 - x^2 = 0.1$$

Az $x = 1, \ldots, 100$ értékekre számítógépén tesztelje a fenti egyenlőség teljesülését!

Kerekítés

Az **abszolút hiba** becslése

szabályos kerekítésnél:

$$|fl(x) - x| \leq \begin{cases} \varepsilon_0, & \text{ha } |x| < \varepsilon_0 \\ \frac{1}{2}\varepsilon_1 |x|, & \text{ha } |x| \geq \varepsilon_0 \end{cases}$$

levágásnál:

$$|fl(x) - x| \leq \begin{cases} \varepsilon_0, & \text{ha } |x| < \varepsilon_0 \\ \varepsilon_1 |x|, & \text{ha } |x| \geq \varepsilon_0 \end{cases}$$

Kerekítés

A **relatív hiba** becslése, ha $|x| \geq \varepsilon_0$

szabályos kerekítésnél:

$$\frac{|fl(x) - x|}{|x|} \leq \frac{1}{2}\varepsilon_1$$

levágásnál:

$$\frac{|fl(x) - x|}{|x|} \leq \varepsilon_1$$

Gépi epszilon (ε_1)

Adott számábrázolási jellemzők mellett az 1 és a jobboldali lebegőpontos szomszédjának a távolsága.

Alapműveleteknél:

1. példa:

$a = 10,\ t = 3$
$x = 0.425 \cdot 10^{-1},\ y = 0.677 \cdot 10^{-2}$

$fl(x + y) = ?$

$y \to y = 0.0677 \cdot 10^{-1}$ (**tartalék számjegyek**)

$x + y = 0.425 \cdot 10^{-1} + 0.0677 \cdot 10^{-1} = 0.4927 \cdot 10^{-1}$

$$fl(x + y) = \begin{cases} 0.492 \cdot 10^{-1}, & \text{levágás} \\ 0.493 \cdot 10^{-1}, & \text{szabályos kerekítés} \end{cases}$$

2. példa:

$a = 10,\ t = 3$
$x = 0.367 \cdot 10^{-2},\ y = 0.682 \cdot 10^{-2}$

$fl(x + y) = ?$

$x + y = 0.367 \cdot 10^{-2} + 0.682 \cdot 10^{-2} = 1.049 \cdot 10^{-2} = 0.1049 \cdot 10^{-1}$

$$fl(x + y) = \begin{cases} 0.104 \cdot 10^{-1}, & \text{levágás} \\ 0.105 \cdot 10^{-1}, & \text{szabályos kerekítés} \end{cases}$$

Alapműveleteknél:

Jelölje $\triangle$ a négy alapművelet valamelyikét, legyen x és y lebegőpontos szám. Tfh a gép a műveletet pontosan végrehajtja és az eredményhez hozzárendel egy lebegőpontos számot. Ekkor

szabályos kerekítés esetén:

$$|fl(x\triangle y) - x\triangle y| \leq \begin{cases} \varepsilon_0, & \text{ha } |x\triangle y| < \varepsilon_0 \\ \frac{1}{2}\varepsilon_1|x\triangle y|, & \text{ha } |x\triangle y| \geq \varepsilon_0 \end{cases}$$

levágás esetén:

$$|fl(x\triangle y) - x\triangle y| \leq \begin{cases} \varepsilon_0, & \text{ha } |x\triangle y| < \varepsilon_0 \\ \varepsilon_1|x\triangle y|, & \text{ha } |x\triangle y| \geq \varepsilon_0 \end{cases}$$

Összefoglalva:

ha $|x \triangle y| > M_\infty$, akkor **túlcsordulás**,

ha $|x \triangle y| < \varepsilon_0$, akkor **alulcsordulás** ($fl(x \triangle y) = 0$)

ha $\varepsilon_0 \leq |x \triangle y| \leq M_\infty$, akkor az előző reláció átírható:

$$fl(x \triangle y) = (x \triangle y) \cdot (1 + \varepsilon_\triangle), \quad \text{ahol } |\varepsilon_\triangle| \leq \varepsilon_1 \begin{cases} 1, & \text{levágás} \\ \frac{1}{2}, & \text{szabályos kerekítés} \end{cases}$$

A hibák terjedése

Legyenek $x_0, x_1, \ldots, x_n$ lebegőpontos számok.

$S_n = \sum_{i=0}^{n} x_i = ?$, ha az összeadás algoritmusa:

$$S_0 = x_0, \qquad S_k = S_{k-1} + x_k, \quad k = 1, \ldots, n.$$

A hiba becslése:

$$|fl(S_n) - S_n| \leq n\varepsilon_1|x_0| + n\varepsilon_1|x_1| + (n-1)\varepsilon_1|x_2| + \cdots + \varepsilon_1|x_n|$$

Egy durvább becslés:

$$|fl(S_n) - S_n| \leq n\varepsilon_1 \sum_{k=0}^{n} |x_k|$$

Ha minden x_k pozitív, akkor

$$\left| \frac{fl(S_n) - S_n}{S_n} \right| \leq n\varepsilon_1$$

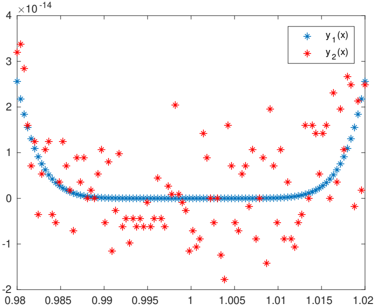
Megjegyzések

- A lebegőpontos összeadás nem asszociatív
- Az elvégzett műveletek számának növekedésével a kerekítési hiba tipikusan nő. Matematikailag ekvivalens kifejezések értékére lényegesen különböző értékeket kaphatunk a gépi számítás során.

Példa (labor)

Számítógépén határozza meg és ábrázolja az 1 egy kis környezetében az $(x-1)^8$ kifejezés értéket az alábbi két (matematikailag ekvivalens) módon:

$$y_1(x) = (x-1)^8,$$
$$y_2(x) = x^8 - 8x^7 + 28x^6 - 56x^5 + 70x^4 - 56x^3 + 28x^2 - 8x + 1$$



Néhány, numerikus probléma okozta katasztrófa

1. Patriot rakéta probléma

1991. február 25-én, az Öböl-háborúban, Dhahran-ban (Szaud-Arábia) az amerikai légvédelmi üteg Patriot rakétája egy iraki Scud rakéta elfogása és követése során célt tévesztett. A Scud rakéta becsapódása 28 katona halálát és további 97 ember sérülését okozta.

A Patriot vezérlő számítógépe az időt $\frac{1}{10}$ másodperces egységekben mérte, az idő másodpercben való visszanyeréséhez egy adott ponton $\frac{1}{10}$-del szoroztak. A gép egyszeres pontosságú számokat használt, ezért az $\frac{1}{10}$ bináris ábrázolása során kb. 0.000000095 hiba keletkezett. Az Scud rakéta érkezésekor a számítógép már 100 órája működött, így a hiba kb. 0.34 másodpercre nőtt. A Scud rakéta sebessége kb 1676 m/s, azaz ennyi idő alatt több, mint 500 m-t tett meg. A rakéta első érzékelése után így az automatika az ég rossz szegletére pillantott, és az első észlelést detektálási hibának minősítette, nem indított elhárító rakétát.

`http://www-users.math.umn.edu/~arnold//disasters/patriot.html`

2. Az Ariane 5 rakéta felrobbanása

1996 június 4-én az Európai Űrügynökség Ariane 5 rakétája kb 40 másodperccel a kilövés után felrobbant. A keletkező anyagi kár kb 500 millió dollár.

A problémát az okozta, hogy egy sebességi értéket 16 bites előjeles egészként akartak tárolni, de az nagyobb volt, mint a legnagyobb így ábrázolható szám.

http://www-users.math.umn.edu/~arnold//disasters/ariane.html
http://www-users.math.umn.edu/~arnold//disasters/ariane5rep.html