Teoria erorilor și aritmetică în virgulă flotantă

Erorile sunt omniprezente

Radu Tiberiu Trîmbițaș

Universitatea "Babeș-Bolyai"

Februarie 2010

Tipuri de erori

Aprecierea preciziei rezultatelor calculelor este un obiectiv important în Analiza numerică. Se disting mai multe tipuri de erori care pot limita această precizie:

1. erori în datele de intrare - sunt în afara (dincolo de) controlului calculelor. Ele se pot datora, de exemplu, imperfecțiunilor inerente ale măsurătorilor fizice.
2. erori de rotunjire - apar dacă se fac calcule cu numere a căror reprezentare se restrânge la un număr finit de cifre.
3. erori de aproximare -multe metode nu dau soluția exactă a problemei P, ci a unei probleme mai simple $\widetilde{P}$, care aproximează P: integralele se aproximează prin sume finite, derivatele prin diferențe (divizate), etc. Aceste erori se numesc erori de discretizare.

Exemplu de eroare de aproximare

- (P) Dorim să aproximăm

$$e = 1 + \frac{1}{1!} + \cdots + \frac{1}{n!} + \ldots$$

- Problema se înlocuiește cu problema mai simplă $(\widetilde{P})$ a însumării unui număr finit de termeni -eroare se trunchiere

$$(\widetilde{P}) \qquad e = 1 + \frac{1}{1!} + \cdots + \frac{1}{n!}.$$

- În acest capitol ne interesează doar erorile în datele de intrare și erorile de rotunjire.

Probleme numerice

- Combinația dintre o problemă matematică (PM), (de natură constructivă) și specificațiile de precizie ale rezultatului (SP) se numește problemă numerică.

Probleme numerice

- Combinația dintre o problemă matematică (PM), (de natură constructivă) și specificațiile de precizie ale rezultatului (SP) se numește problemă numerică.
- Exemplu: Fie $f : \mathbb{R} \longrightarrow \mathbb{R}$ și $x \in \mathbb{R}$. Dorim să calculăm $y = f(x)$. În general x nu este reprezentabil în calculator; din acest motiv vom lucra cu o aproximare x^* a sa, $x^* \approx x$. De asemenea este posibil ca f să nu poată fi calculată exact; vom înlocui f cu o aproximantă a sa f_A. Valoarea calculată în calculator va fi $f_A(x^*)$. Deci problema numerică este următoarea:

Probleme numerice

- Combinația dintre o problemă matematică (PM), (de natură constructivă) și specificațiile de precizie ale rezultatului (SP) se numește problemă numerică.
- Exemplu: Fie $f : \mathbb{R} \longrightarrow \mathbb{R}$ și $x \in \mathbb{R}$. Dorim să calculăm $y = f(x)$. În general x nu este reprezentabil în calculator; din acest motiv vom lucra cu o aproximare x^* a sa, $x^* \approx x$. De asemenea este posibil ca f să nu poată fi calculată exact; vom înlocui f cu o aproximantă a sa f_A. Valoarea calculată în calculator va fi $f_A(x^*)$. Deci problema numerică este următoarea:

  PM. dându-se x și f, să se calculeze $f(x)$;

Probleme numerice

- Combinația dintre o problemă matematică (PM), (de natură constructivă) și specificațiile de precizie ale rezultatului (SP) se numește problemă numerică.
- Exemplu: Fie $f : \mathbb{R} \longrightarrow \mathbb{R}$ și $x \in \mathbb{R}$. Dorim să calculăm $y = f(x)$. În general x nu este reprezentabil în calculator; din acest motiv vom lucra cu o aproximare x^* a sa, $x^* \approx x$. De asemenea este posibil ca f să nu poată fi calculată exact; vom înlocui f cu o aproximantă a sa f_A. Valoarea calculată în calculator va fi $f_A(x^*)$. Deci problema numerică este următoarea:

PM. dându-se x și f, să se calculeze $f(x)$;
SP. $|f(x) - f_A(x^*)| < \varepsilon$, ε dat.

Măsuri ale erorii

- X spațiu liniar normat, $A \subseteq X$, $x \in X$. Un element $x^* \in A$ se numește aproximantă a lui x din A (notație $x^* \approx x$).
- $x^* \approx x$ o aproximantă a lui x, diferența $\Delta x = x - x^*$ se numește eroare, iar
$$\|\Delta x\| = \|x^* - x\| \tag{1}$$
se numește eroare absolută.
- $$\delta x = \frac{\|\Delta x\|}{\|x\|}, \quad x \neq 0 \tag{2}$$
se numește eroare relativă.
- Deoarece în practică x este necunoscut, se folosește aproximarea $\delta x = \frac{\|\Delta x\|}{\|x^*\|}$. Dacă $\|\Delta x\|$ este mic comparativ cu $\|x^*\|$, atunci aproximanta este bună.

Eroarea propagată

- $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $x = (x_1, \dots, x_n)$, $x^* = (x_1^*, \dots, x_n^*)$. Dorim să evaluăm eroarea absolută și relativă Δf și respectiv δf când se aproximează $f(x)$ prin $f(x^*)$.
- Aceste erori se numesc erori propagate, deoarece ne spun cum se propagă eroarea inițială (absolută sau relativă) pe parcursul calculării lui f.
- Presupunem $x = x^* + \Delta x$, $\Delta x = (\Delta x_1, \dots, \Delta x_n)$. Aplicăm formula lui Taylor

$$\begin{aligned}\Delta f &= f(x_1^* + \Delta x_1, \dots, x_n^* + \Delta x_n) - f(x_1^*, \dots x_n^*)\\ &= \sum_{i=1}^{n} \Delta x_i \frac{\partial f}{\partial x_i^*}(x_1^*, \dots x_n^*) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \Delta x_i \Delta x_j \frac{\partial^2 f}{\partial x_i^* \partial x_j^*}(\theta),\end{aligned}$$

$\theta \in [(x_1^*, \dots, x_n^*), (x_1^* + \Delta x_1, \dots, x_n^* + \Delta x_n)]$.

Eroarea propagată II

- neglijând termenii de ordinul al doilea (mici) obținem

$$\Delta f \approx \sum_{i=1}^{n} \Delta x_i \frac{\partial f}{\partial x_i^*}(x_1^*, \ldots x_n^*). \tag{3}$$

- Pentru eroarea relativă avem

$$\begin{aligned} \delta f = \frac{\Delta f}{f} &\approx \sum_{i=1}^{n} \Delta x_i \frac{\frac{\partial f}{\partial x_i^*}(x^*)}{f(x^*)} \\ &= \sum_{i=1}^{n} \delta x_i \frac{x_i^* \frac{\partial f}{\partial x_i^*}(x^*)}{f(x^*)} \end{aligned} \tag{4}$$

Eroarea propagată III

- Problema inversă: cu ce precizie trebuie aproximate datele pentru ca rezultatul să aibă o precizie dată?
- Adică, dându-se $\varepsilon > 0$, cât trebuie să fie Δx_i sau δx_i, $i = \overline{1, n}$ astfel încât Δf sau $\delta f < \varepsilon$?
- principiul efectelor egale: se presupune că toți termenii care intervin în (3) sau (4) au același efect, adică

$$\frac{\partial f}{\partial x_1^*}(x^*)\Delta x_1 = \ldots = \frac{\partial f}{\partial x_n^*}(x^*)\Delta x_n.$$

- se obține

$$\Delta x_i \approx \frac{\Delta f}{n\left|\frac{\partial f}{\partial x_i^*}(x^*)\right|}. \tag{5}$$

$$\delta x_i = \frac{\delta f}{n\left|\frac{x_i^* \frac{\partial}{\partial x_i^*} f(x^*)}{f(x^*)}\right|}. \tag{6}$$

Exemple

Exemplu. Găsiți o margine a erorii absolute și relative pentru volumul sferei $V = \frac{\pi d^3}{6}$ cu diametrul egal cu $3.7cm \pm 0.04cm$ și $\pi \approx 3.14$.

- Calculăm derivatele parțiale

$$\frac{\partial V}{\partial \pi} = \frac{1}{6}d^3 = 8.44, \qquad \frac{\partial V}{\partial d} = \frac{1}{2}\pi d^2 = 21.5.$$

- Aplicând formula (3) și definiția erorii relative obținem:

$$\Delta V = \left|\frac{\partial V}{\partial \pi}\right| |\Delta \pi| + \left|\frac{\partial V}{\partial d}\right| |\Delta d| = 8.44 + 21.5 \cdot 0.05 \approx 1.0888,$$

$$\delta_V = \frac{1.0888}{274} \approx 4\%.$$

Exemple - continuare

Exemplu. Un cilindru are raza $R \approx 2m$, înălținea $H \approx 3m$. Cu ce erori absolute trebuie determinate R, H și π astfel încât V să poată fi calculat cu o eroare $< 0.1m^3$.
Se aplică principiul efectelor egale (5):

$$V = \pi R^2 H, \quad \Delta V = 0.1m^3,$$

$$\frac{\partial V}{\partial \pi} = R^2 H = 12, \quad \frac{\partial V}{\partial R} = 2\pi R H = 37.7, \quad \frac{\partial V}{\partial H} = \pi R^2 = 12.6.$$

$n = 3$, erorile absolute ale argumentelor:

$$\Delta \pi \approx \frac{\Delta V}{3\dfrac{\partial V}{\partial \pi}} = \frac{0.1}{3.12} < 0.003,$$

$$\Delta R \approx \frac{0.1}{3 \cdot 37.7} < 0.001, \qquad \Delta H \approx \frac{0.1}{3 \cdot 12.6} < 0.003.$$

- Parametrii reprezentării în virgulă flotantă sunt următoarele numere naturale
  - baza β (întotdeauna pară);
  - precizia p;
  - exponentul maxim $e_{\max}$;
  - exponentul minim $e_{\min}$;
- În general, un număr în virgulă flotantă se reprezintă sub forma

$$x = \pm d_0.d_1 d_2 \dots d_{p-1} \times \beta^e, \quad 0 \leq d_i < \beta \tag{7}$$

$d_0.d_1 d_2 \dots d_{p-1}$ - semnificant sau mantisă, e exponent.

- Valoarea lui x este

$$\pm\left(d_0 + d_1\beta^{-1} + d_2\beta^{-2} + \dots + d_{p-1}\beta^{-(p-1)}\right)\beta^e. \tag{8}$$

Parametrii reprezentării

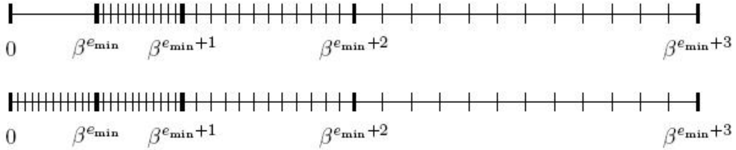
- Unicitatea se asigură prin normalizare: se modifică reprezentarea (nu valoarea) astfel încât $d_0 \neq 0$.
- Zero se reprezintă ca $1.0 \times \beta^{e_{\min}-1}$
- Ordinea numerică uzuală a numerelor reale nenegative corespunde ordinii lexicografice a reprezentării lor flotante (cu exponentul în stânga semnificantului).
- număr în virgulă flotantă = număr real care poate fi reprezentat exact în virgulă flotantă

Numere denormalizate

- Dupa normalizarea semnificanților ramâne un ,,gol" între 0 și $\beta^{e_{min}}$
- Aceasta poate avea ca efect $x - y = 0$ chiar dacă $x \neq y$, iar un fragment de cod de tipul **if** $x \neq y$ **then** $z = 1/(x - y)$ poate eșua
- Soluție: se admit semnificanți nenormalizați când exponentul este $e_{\min}$ (gradual underflow). Aceste numere se numesc numere denormalizate. Ele garantează că

$$x = y \iff x - y = 0$$

- Distribuția fără denormalizare și cu denormalizare



Parametrii reprezentării

- Mulțimea numerelor în virgulă flotantă pentru un set de parametri dați ai reprezentării se va nota cu

$$\mathbb{F}(\beta, p, e_{\min}, e_{\max}, \mathrm{denorm}), \quad \mathrm{denorm} \in \{\mathit{true}, \mathit{false}\}.$$

- Această mulțime nu coincide cu $\mathbb{R}$ din următoarele motive:
  1. este o submulțime finită a lui $\mathbb{Q}$;
  2. pentru $x \in \mathbb{R}$ putem avea $|x| > \beta \times \beta^{e_{\max}}$ (depășire superioară) sau $|x| < 1.0 \times \beta^{e_{\min}}$ (depășire inferioară).
- Operațiile aritmetice uzuale pe $\mathbb{F}(\beta, p, e_{\min}, e_{\max}, \mathrm{denorm})$ se notează cu $\oplus$, $\ominus$, $\otimes$, $\oslash$, iar funcțiile uzuale cu SIN, COS, EXP, LN, SQRT ș.a.m.d. $(\mathbb{F}, \oplus, \otimes)$ nu este corp deoarece

$$(x \oplus y) \oplus z \neq x \oplus (y \oplus z) \qquad (x \otimes y) \otimes z \neq x \otimes (y \otimes z)$$
$$(x \oplus y) \otimes z \neq x \otimes z \oplus y \otimes z.$$

Erori

- Eroarea relativă
- ulps – units in the last place (unități în ultima poziție): dacă $z = d_0.d_1d_2\ldots d_{p-1}\ldots \times \beta^e$, atunci eroarea este

$$\left|d_0.d_1d_2\ldots d_{p-1} - z/\beta^e\right| \beta^{p-1}\text{ulps}.$$

- Eroarea relativă ce corespunde la $\frac{1}{2}$ulps este

$$\frac{1}{2}\beta^{-p} \leq \frac{1}{2}\text{ulps} \leq \frac{\beta}{2}\beta^{-p},$$

căci eroarea absolută este $\underbrace{0.0\ldots0}_{p}\beta' \times \beta^e$, cu $\beta' = \frac{\beta}{2}$. Valoarea $\text{eps} = \frac{\beta}{2}\beta^{-p}$ se numește epsilon-ul mașinii.

- Echivalent rezoluția relativă (distanța relativă între doi vecini)

Rotunjire

- Rotunjirea implicită se face după regula cifrei pare: dacă $x = d_0.d_1 \ldots d_{p-1} d_p \ldots$ și $d_p > \frac{\beta}{2}$ rotunjirea se face în sus, dacă $d_p < \frac{\beta}{2}$ rotunjirea se face în jos, iar dacă $d_p = \frac{\beta}{2}$ și printre cifrele eliminate există una nenulă rotunjirea se face în sus, iar în caz contrar ultima cifră păstrată este pară.
- Alte tipuri de rotunjiri: în jos, în sus, spre zero, trunchiere

Aritmetică în virgulă flotantă

- Definim $\mathrm{fl}(x)$ ca fiind cea mai apropiată aproximare în virgulă flotantă a lui x
- Din definiția eps avem pentru eroarea relativă:

$$\forall x \in \mathbb{R},\ \exists \epsilon \text{ cu } |\epsilon| \leq \text{eps astfel încât } \mathrm{fl}(x) = x(1+\epsilon)$$

- Rezultatul unei operații $\circledcirc$ în virgulă flotantă este $\mathrm{fl}(a \circ b)$
- Dacă $\mathrm{fl}(a \circ b)$ este cel mai apropiat număr în virgulă flotantă de $a \circ b$, operațiile aritmetice se rotunjesc corect (standardul IEEE o face), ceea ce ne conduce la următoarea proprietate:

  Pentru orice numere în virgulă flotantă x, y, există ϵ cu $|\epsilon| \leq$ eps astfel încât

$$x \circledcirc y = (x \circ y)(1+\epsilon)$$

  numită axioma fundamentală a aritmeticii în virgulă flotantă
- Rotunjire la cel mai apropiat par în caz de ambiguitate

Anularea

- Din formulele pentru eroarea relativă (4), dacă $x \approx x(1+\delta_x)$ și $y \approx y(1+\delta_y)$, avem următoarele expresii pentru erorile relative ale operațiilor în virgulă flotantă:

$$\delta_{xy} = \delta_x + \delta_y \tag{9}$$

$$\delta_{x/y} = \delta_x - \delta_y \tag{10}$$

$$\delta_{x+y} = \frac{x}{x+y}\delta_x + \frac{y}{x+y}\delta_y \tag{11}$$

- Singura operație critică din punct de vedere al erorii este scăderea a două cantități apropiate $x \approx y$, caz în care $\delta_{x-y} \to \infty$.
- Acest fenomen se numește anulare

Anularea II

- Anularea este de două tipuri:
    1. benignă, când se scad două cantități exacte
    2. catastrofală, când se scad două cantități deja rotunjite.
- Programatorul trebuie să fie conștient de posibilitatea apariției anulării și să încerce să o evite.
- Expresiile în care apare anularea trebuie rescrise, iar o anulare catastrofală trebuie întotdeauna transformată în una benignă.

Anularea III

- **Exemplu.** Dacă $a \approx b$, atunci expresia $a^2 - b^2$ se transformă în $(a-b)(a+b)$. Forma inițială este de preferat în cazul când $a \gg b$ sau $b \gg a$.
- **Exemplu.** Dacă anularea apare într-o expresie cu radicali, se amplifică cu conjugata:

$$\sqrt{x+\delta} - \sqrt{x} = \frac{\delta}{\sqrt{x+\delta} + \sqrt{x}}, \quad \delta \approx 0.$$

- **Exemplu.** Diferența valorilor unei funcții pentru argumente apropiate se transformă folosind formula lui Taylor:

$$f(x+\delta) - f(x) = \delta f'(x) + \frac{\delta^2}{2} f''(x) + \cdots \quad f \in C^n[a,b].$$

Anularea IV

La ecuația de gradul al doilea $ax^2 + bx + c = 0$, anularea poate să apară dacă $b^2 \gg 4ac$. Formulele uzuale

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{12}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{13}$$

pot să conducă la anulare astfel: pentru $b > 0$ anularea apare la calculul lui x_1, iar pentru $b < 0$ anularea apare la calculul lui x_2. Remediul este să amplificăm cu conjugata

$$x_1 = \frac{2c}{-b - \sqrt{b^2 - 4ac}} \tag{14}$$

$$x_2 = \frac{2c}{-b + \sqrt{b^2 - 4ac}} \tag{15}$$

și să utilizăm în primul caz formulele (14) și (13), iar în al doilea caz (12) și (15). `../demo/html/ecgr2.html`

Teorema asupra pierderii preciziei

- Problemă: *Câte cifre semnificative se pierd la scăderea $x - y$ când x este apropiat de y?*
- Apropierea lui x de y este măsurată convenabil de $1 - \frac{y}{x}$.

Teoremă (Loss of precission theorem)

Fie x și y NVF normalizate, unde $x > y > 0$. Dacă

$$2^{-p} \leq 1 - \frac{y}{x} \leq 2^{-q}$$

pentru $p, q \in \mathbb{N}$, atunci se pierd cel puțin q și cel mult p cifre binare semnificative la scăderea $x - y$.

Teorema asupra pierderii preciziei - demonstrație

Demonstrație.

Vom demonstra partea a doua, lăsând prima parte ca exercițiu. Fie $x = s \times 2^n$, $y = r \times 2^m$ NVF normalizate. Deoarece $y < x$, y va trebui deplasat înaintea scăderii, pentru a avea același exponent ca x. Deci, $y = \left(s2^{m-n}\right) \times 2^n$ și

$$x - y = \left(r - s2^{m-n}\right) \times 2^n$$

Semnificantul satisface

$$r - s2^{m-n} = r\left(1 - \frac{r \times 2^m}{s \times 2^n}\right) = r\left(1 - \frac{y}{x}\right) < 2^{-q}$$

Deci, pentru normalizarea reprezentării lui $x - y$, este nevoie de o deplasare de q biți la stânga. Astfel se introduc cel puțin q zerouri false la capătul drept al semnificantului. Aceasta înseamnă o pierdere a preciziei de cel puțin q biți. $\square$

Reducerea rangului I

Exemplu

Pentru $\sin x$*, câți biți semnificativi se pierd la reducerea la intervalul* $[0, 2\pi)$*?*

Soluție. Dându-se $x > 2\pi$, vom determina întregul n ce satisface $0 \leq x - 2n\pi < 2\pi$. Apoi la evaluare vom utiliza periodicitatea $f(x) = f(x - 2n\pi)$. La scăderea $x - 2n\pi$, va fi o pierdere de precizie. Conform teoremei 1 se vor pierde cel puțin q biți dacă

$$1 - \frac{2n\pi}{x} \leq 2^{-q}$$

Deoarece

$$1 - \frac{2n\pi}{x} = \frac{x - 2n\pi}{x} < \frac{2\pi}{x}$$

conchidem că cel puțin q biți se pierd dacă $2\pi/x < 2^{-q}$, sau echivalent, dacă $2^q < x/2\pi$. ■

Exemplu numeric. Să se calculeze $\sin(12532.14)$.
Avem $\sin(12532.14) = \sin(12532.14 - 2k\pi)$, cu $k = 1994$ și $12532.14 - 2k\pi \approx 3.47$ și rezultatul va fi eronat. Dacă reducerea s-ar fi putut face cu precizie mai bună și rezultatul ar fi fost mai bun. MATLAB dă $\sin(12532.14) = -0.321113319309938$ și $\sin(3.47) = -0.322535900322479$.

Standardele IEEE

- Există două standarde diferite pentru calculul în virgulă flotantă:
  1. IEEE 754 care prevede $\beta = 2$
  2. IEEE 854 care permite $\beta = 2$ sau $\beta = 10$, dar lasă o mai mare libertate de reprezentare.
- Parametrii standardului IEEE 754

	Precizia			
	Simplă	Simplă extinsă	Dublă	Dublă extinsă
p	24	≥ 32	53	≥ 64
$e_{\max}$	$+127$	$\geq +1023$	$+1023$	$\geq +16383$
$e_{\min}$	-126	≤ -1022	-1022	≤ -16382
dim. exponent	8	≥ 11	11	≥ 15
dim. număr	32	≥ 43	64	≥ 79

Tabela: Parametrii reprezentării flotante

bit ascuns - $d_0 = 1$, deci nu trebuie reprezentat fizic

IEEE 754 II

Motivele pentru formatele extinse sunt:

1. o mai bună precizie;
2. pentru conversia din binar în zecimal și invers este nevoie de 9 cifre în simplă precizie și de 17 cifre în dublă precizie.

Motivul pentru care $|e_{min}| < e_{max}$ este acela că $1/2^{e_{min}}$ nu trebuie să dea depășire.

Operațiile $\oplus, \ominus, \otimes, \oslash$ trebuie să fie exact rotunjite. Precizia aceasta se asigură cu două cifre de gardă și un bit suplimentar.

Reprezentarea exponentului se numește reprezentare cu exponent deplasat, adică în loc de e se reprezintă $e + D$, unde D este fixat la alegerea reprezentării.

$D = 127$ pentru simplă precizie și $D = 1023$ pentru dublă precizie.

Cantități speciale

Exponent	Semnificant	Ce reprezintă	
$e = e_{min} - 1$	$f = 0$	± 0	zero cu semn
$e = e_{min} - 1$	$f \neq 0$	$0.f \times 2^{e_{min}}$	Numere denormalizate
$e_{min} \leq e \leq e_{max}$		$1.f \times 2^{e}$	
$e = e_{max} + 1$	$f = 0$	$\pm\infty$	infinit
$e = e_{max} + 1$	$f \neq 0$	NaN	NaN-uri

Cantități speciale

NaN. Avem de fapt o familie de valori NaN, operațiile ilegale sau nedeterminate conduc la NaN: $\infty + (-\infty)$, $0 \times \infty$, $0/0$, ∞/∞, $x \,\mathrm{REM}\, 0$, $\infty \,\mathrm{REM}\, y$, $\sqrt{x}$ pentru $x < 0$. Dacă un operand este NaN rezultatul va fi tot NaN.

Infinit. Operațiile cu ∞ se definesc ca limite, ex: $1/0 = \infty$, $-1/0 = -\infty$. Valorile infinite dau posibilitatea continuării calculului, lucru mai sigur decât abortarea sau returnarea celui mai mare număr reprezentabil. $\frac{x}{1+x^2}$ pentru $x = \infty$ dă rezultatul 0.

Zero cu semn. Avem doi de 0: $+0, -0$; relațiile $+0 = -0$ și $-0 < +\infty$ sunt adevărate. Avantaje: tratarea simplă a depășirilor inferioare și discontinuităților. Se face distincție între $\log 0 = -\infty$ și $\log x = \mathrm{NaN}$ pentru $x < 0$. Fără 0 cu semn nu s-ar putea face distincție la logaritm între un număr negativ care dă depășire superioară și 0.

IEEE Simplă precizie, exemple

S	E	M	Cantitate
0	11111111	00000100000000000000000	NaN
1	11111111	00100010000100101010101	NaN
0	11111111	00000000000000000000000	∞
0	10000001	10100000000000000000000	$+2^{129-127} \cdot 1.101 = 6.5$
0	10000000	00000000000000000000000	$+2^{128-127} \cdot 1.0 = 2$
0	00000001	00000000000000000000000	$+2^{1-127} \cdot 1.0 = 2^{-126}$
0	00000000	10000000000000000000000	$+2^{-126} \cdot 0.1 = 2^{-127}$
0	00000000	00000000000000000000001	$+2^{-126} \cdot 2^{-23} = 2^{-149}$
0	00000000	00000000000000000000000	$+0$
1	00000000	00000000000000000000000	-0
1	10000001	10100000000000000000000	$-2^{129-127} \cdot 1.101 = -6.5$
1	11111111	00000000000000000000000	$-\infty$

Pentru virgulă flotantă în MATLAB vezi `../demo/html/fpdemo.html`

William Kahan, eminent matematician și informatician, contribuții importante la studiul metodelor precise și eficiente de rezolvare a problemelor numerice pe calculatoare cu precizie finită. A fost principalul arhitect al standardului IEEE 754. Distins cu premiul Turing al ACM în 1989, Fellow al ACM din 1994. Profesor la Universitatea Berkeley, California

Eșecul rachetei Patriot I

- Eșecul unui sistem de rachete antirachetă Patriot în timpul războiului din Golf din 1991 s-a datorat unei erori de conversie software.
- Ceasul sistemului măsura timpul în zecimi de secundă, dar îl memora într-un registru de 24 de biți, provocându-se astfel erori de rotunjire.
- Datele din câmp au arătat că sistemul poate eșua să urmărească și să intercepteze o rachetă după 20 de ore de funcționare și deci sistemul ar necesita rebootare.

Eșecul rachetei Patriot II

- După 100 de ore de funcționare, eșecul sistemului a cauzat moartea a 28 de soldați americani aflați într-o cazarmă din Dhahran, Arabia Saudită, deoarece nu a reușit să intercepteze o rachetă Scud irakiană. Deoarece numărul 0.1 are o dezvoltare infinită în binar (este o fracție periodică), valoare din registrul de 24 de biți este eronată

$$(0.00011001100110011001100)_2 \approx 0.95 \times 10^{-7}.$$

Eroarea de timp după o sută de ore a fost de 0.34 secunde. Viteza rachetei Scud este de 3750 mile/oră, rezultând o eroare în distanță de aproximativ 573.59 m.

Explozia rachetei Ariane 5

- În 1996, racheta Ariane 5 lansată de Agenția Spațială European a explodat la 40 de secunde după lansarea de la Kourou, Guyana Franceză.
- Investigația de după incident a arătat că componeta orizontală a vitezei a necesitat conversia unui număr flotant în dublă precizie într-un întreg pe 16 biți.
- Deoarece numărul era mai mare decât 32,767, cel mai mare întreg reprezentabil pe 16 biți, componentele de control au intrat în procedura de autodistrugere. Valoarea rachetei și a încărcăturii a fost de 500 de milioane de dolari.

Referințe WWW

Se pot găsi informații adiționale pe World Wide Web la adresa http://www.ima.umn.edu/~arnold/disasters/ sau la http://www5.in.tum.de/~huckle/bugse.html. Există și alte consemnări ale calamităților ce ar fi putut fi evitate printr-o programare mai atentă, în special la utilizarea AVF.

- Putem gândi o problemă ca o aplicație

$$f : \mathbb{R}^m \to \mathbb{R}^n, \quad y = f(x). \tag{16}$$

- Ne interesează sensibilitatea aplicației într-un punct dat x la mici perturbații ale argumentului, adică cât de mare sau cât de mică este perturbația lui y comparativ cu perturbația lui x.
- În particular, dorim să măsurăm gradul de sensibilitate printr-un singur număr, numărul de condiționare al aplicației f în punctul x. Vom presupune că f este calculată exact, cu precizie infinită.
- Condiționarea lui f este deci o proprietate inerentă a funcției f și nu depinde de nici o considerație algoritmică legată de implementarea sa.

Condiționarea unei probleme

- Aceasta nu înseamnă că determinarea condiționării unei probleme este nerelevantă pentru orice soluție algoritmică a problemei.
- Soluția calculată cu (16), y^* (utilizând un algoritm specific și aritmetica în virgulă flotantă) este (și acest lucru se poate demonstra) soluția unei probleme ,,apropiate"

$$y^* = f(x^*) \tag{17}$$

cu

$$x^* = x + \delta \tag{18}$$

- distanța $\|\delta\| = \|x^* - x\|$ poate fi estimată în termeni de precizie a mașinii
- dacă știm cât de tare sau cât de slab reacționează aplicația la mici perturbații, cum ar fi δ în (18), putem spune ceva despre eroarea $y^* - y$ a soluției cauzată de această perturbație.

Fie $x = [x_1, \dots, x_m]^T \in \mathbb{R}^m$, $y = [y_1, \dots, y_n]^T \in \mathbb{R}^n$, $y_\nu = f_\nu(x_1, \dots, x_m)$, $\nu = \overline{1, n}$ — y_ν va fi privit ca o funcție de o singură variabilă x_μ

$$\gamma_{\nu\mu} = (\mathrm{cond}_{\nu\mu}\, f)(x) = \left| \frac{x_\mu \frac{\partial f_\nu}{\partial x_\mu}}{f_\nu(x)} \right|. \tag{19}$$

Aceasta ne dă o matrice de numere de condiționare

$$\Gamma(x) = \begin{pmatrix} \frac{x_1 \frac{\partial f_1}{\partial x_1}}{f_1(x)} & \dots & \frac{x_m \frac{\partial f_1}{\partial x_m}}{f_1(x)} \\ \vdots & \ddots & \vdots \\ \frac{x_1 \frac{\partial f_n}{\partial x_1}}{f_n(x)} & \dots & \frac{x_m \frac{\partial f_n}{\partial x_m}}{f_n(x)} \end{pmatrix} =: [\gamma_{\nu\mu}(x)] \tag{20}$$

și vom lua ca număr de condiționare

$$(\mathrm{cond}\ f)(x) = \|\Gamma(x)\|. \tag{21}$$

Altfel.

$$\|\Delta y\| = \|f(x+\Delta x) - f(x)\| \leq \|\Delta x\| \|\frac{\partial f}{\partial x}\|$$

unde

$$J(x) = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^n \times \mathbb{R}^m \tag{22}$$

este matricea jacobiană a lui f

$$\frac{\|\Delta y\|_\infty}{\|y\|_\infty} \leq \frac{\|x\|_\infty \left\|\frac{\partial f}{\partial x}\right\|_\infty}{\|f(x)\|_\infty} \cdot \frac{\|\Delta x\|}{\|x\|_\infty}. \tag{23}$$

Cazul unidimensional

- Pentru $m = n = 1$ și $x \neq 0$, $y \neq 0$

$$(\operatorname{cond} f)(x) = \left| \frac{x f'(x)}{f(x)} \right|.$$

- Dacă $x = 0 \wedge y \neq 0$ se consideră eroarea absolută pentru x și eroarea relativă pentru y

$$(\operatorname{cond} f)(x) = \left| \frac{f'(x)}{f(x)} \right|;$$

- Pentru $y = 0 \wedge x \neq 0$ se ia eroarea absolută pentru y și eroarea relativă pentru x
- Pentru $x = y = 0$, se iau erorile absolute

$$(\operatorname{cond} f)(x) = f'(x).$$

- Număr de condiționare absolută al unei probleme diferențiabile f în x:

$$\hat{\kappa} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|} = \|J(x)\|$$

unde $J(x) = [J_{ij}] = [\partial f_i / \partial x_j]$, este jacobianul, iar norma este indusă de normele lui δf și δx

- în cazul unidimensional

$$\hat{\kappa} = |f'(x)|.$$

Exemple

- **Exemplu**: Funcția $f(x) = \alpha x$
  - număr de condiționare absolută $\hat{\kappa} = \|J\| = \alpha$
  - număr de condiționare relativă $(\text{cond } f)(x) = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{\alpha}{\alpha x/x} = 1$
- **Exemplu**: Funcția $f(x) = \sqrt{x}$
  - număr de condiționare absolută $\hat{\kappa} = \|J\| = \frac{1}{2\sqrt{x}}$
  - număr de condiționare relativă $(\text{cond } f)(x) = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{1/(2\sqrt{x})}{\sqrt{x}/x} = \frac{1}{2}$
- **Exemplu**: Funcția $f(x) = x_1 - x_2$ (cu norma ∞)
  - număr de condiționare absolută $\hat{\kappa} = \|J\| = \left\|(-1, 1)^T\right\| = 2$
  - număr de condiționare relativă

$$(\text{cond } f)(x) = \frac{\|J\|}{\|f(x)\| \;/\; \|x\|} = \frac{2}{|x_1 - x_2| \max\{|x_1|, |x_2|\}}$$

  - prost condiționată dacă $x_1 \approx x_2$ (anulare)

Precizia

- Pentru o funcție dată $g(n)$ vom nota cu $\Theta(g(n))$ mulțimea de funcții

$$\Theta(g(n)) = \{f(n) : \ \exists c_1, c_2, n_0 > 0 \ \ 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \\ \forall n \geq n_0\} .$$

- Scriem $f(n) = \Theta(g(n))$ pentru a indica $f(n) \in \Theta(g(n))$. Spunem că $g(n)$ este o *margine asimptotică strânsă (assymptotically tight bound)* pentru $f(n)$.
- Definiția mulțimii $\Theta(g(n))$ necesită ca fiecare membru al ei să fie *asimptotic nenegativ*, adică $f(n) \geq 0$ când n este suficient de mare.

Precizia

- Pentru o funcție dată $g(n)$ vom nota cu $O(g(n))$ mulțimea de funcții

$$O(g(n)) = \{f(n) : \exists c, n_0 \; 0 \leq f(n) \leq cg(n), \; \forall n \geq n_0\}.$$

- *margine asimptotică superioară*
- Pentru a indica faptul că $f(n)$ este un membru al lui $O(g(n))$ scriem $f(n) = O(g(n))$.
- Observăm că $f(n) = \Theta(g(n)) \implies f(n) = O(g(n)$, sau $\Theta(g(n)) \subseteq O(g(n))$
- Una dintre proprietățile ciudate ale notației este aceea că $n = O(n^2)$.

Precizia

- Pentru o funcție dată $g(n)$ vom nota prin $\Omega(g(n))$ mulțimea de funcții

$$\Omega(g(n)) = \{f(n) : \ \exists c, n_0 \ 0 \leq cg(n) \leq f(n), \ \forall n \geq n_0\}.$$

- *margine asimptotică inferioară*
- Din definițiile notațiilor asimptotice se obține imediat:

$$f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \wedge f(n) = \Omega(g(n)).$$

Precizia

- Spunem că funcțiile f și $g : \mathbb{N} \longrightarrow \mathbb{R}$ sunt *asimptotic echivalente*, notație $\sim$ dacă
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1.$$
- Extinderea notațiilor asimptotice la mulțimea numerelor reale este naturală. De exemplu $f(t) = O(g(t))$ înseamnă că există o constantă pozitivă C astfel încât pentru orice t suficient de apropiat de o limită subînțeleasă (de exemplu $t \to \infty$ sau $t \to 0$) avem
$$|f(t)| \leq Cg(t). \tag{24}$$

Precizia

- Considerăm un *algoritm* $\widetilde{f}$ pentru *problema* f
- Un calcul $\widetilde{f}(x)$ are *eroarea absolută* $\left\|\widetilde{f}(x) - f(x)\right\|$ și *eroarea relativă*

$$\frac{\left\|\widetilde{f}(x) - f(x)\right\|}{\|f(x)\|}$$

- Algoritmul este precis dacă (pentru orice x)

$$\frac{\left\|\widetilde{f}(x) - f(x)\right\|}{\|f(x)\|} = O(\mathrm{eps})$$

  unde $O(\mathrm{eps})$ este “de ordinul eps” (vezi slide-ul următor)
- Constanta din $O(\mathrm{eps})$ poate fi foarte mare pentru multe probleme, căci datorită erorilor de rotunjire nu utilizăm nici chiar un x corect.

Detalii asupra notațiilor asimptotice

- Notația $\varphi(t) = O(\psi(t))$ înseamnă că există o constantă C a.î. pentru t apropiat de o limită (de obicei 0 sau ∞), $|\varphi(t)| \leq C\psi(t)$
- **Exemplu**: $\sin^2 t = O(t^2)$ când $t \to 0$ înseamnă $|\sin^2 t| \leq Ct^2$ pentru un anumit C
- Dacă φ depinde de variabile adiționale, notația

$$\varphi(s,t) = O(\psi(t)) \qquad \text{uniform în } s$$

  înseamnă că există o constantă C a.î. $|\varphi(s,t)| \leq C\psi(t)$ pentru orice s
- **Exemplu**: $(\sin^2 t)(\sin^2 s) = O(t^2)$ uniform când $t \to 0$, dar nu dacă $\sin^2 s$ este înlocuit cu s^2
- În margini de forma $\|\widetilde{x} - x\| \leq C\kappa(A)\text{eps}\,\|x\|$, C nu depinde de de A sau b, dar poate depinde de dimensiunea m

Stabilitatea

- Un algoritm $\widetilde{f}$ pentru problema f este stabil dacă pentru orice x există un $\widetilde{x}$ cu proprietatea

$$\frac{\|\widetilde{x}-x\|}{\|x\|} = O(\mathrm{eps})$$

a.î.

$$\frac{\left\|\widetilde{f}(x)-f(x)\right\|}{\|f(x)\|} = O(\mathrm{eps})$$

Stabilitatea

- Un algoritm $\widetilde{f}$ pentru problema f este stabil dacă pentru orice x există un $\widetilde{x}$ cu proprietatea
$$\frac{\|\widetilde{x}-x\|}{\|x\|}=O(\mathrm{eps})$$
a.î.
$$\frac{\left\|\widetilde{f}(x)-f(x)\right\|}{\|f(x)\|}=O(\mathrm{eps})$$
- "Răspuns aproape corect la problemă aproape exactă"

Stabilitatea

- Un algoritm $\widetilde{f}$ pentru problema f este stabil dacă pentru orice x există un $\widetilde{x}$ cu proprietatea
$$\frac{\|\widetilde{x}-x\|}{\|x\|}=O(\mathrm{eps})$$
a.î.
$$\frac{\left\|\widetilde{f}(x)-f(x)\right\|}{\|f(x)\|}=O(\mathrm{eps})$$
- "Răspuns aproape corect la problemă aproape exactă"
- Un algoritm $\widetilde{f}$ pentru problema f este regresiv stabil dacă pentru orice x există un $\widetilde{x}$ cu proprietatea
$$\frac{\|\widetilde{x}-x\|}{\|x\|}=O(\mathrm{eps})$$
a.î.
$$\widetilde{f}(x)=f(\widetilde{x}).$$

Stabilitatea

- Un algoritm $\widetilde{f}$ pentru problema f este stabil dacă pentru orice x există un $\widetilde{x}$ cu proprietatea
$$\frac{\|\widetilde{x}-x\|}{\|x\|}=O(\mathrm{eps})$$
a.î.
$$\frac{\left\|\widetilde{f}(x)-f(x)\right\|}{\|f(x)\|}=O(\mathrm{eps})$$
- "Răspuns aproape corect la problemă aproape exactă"
- Un algoritm $\widetilde{f}$ pentru problema f este regresiv stabil dacă pentru orice x există un $\widetilde{x}$ cu proprietatea
$$\frac{\|\widetilde{x}-x\|}{\|x\|}=O(\mathrm{eps})$$
a.î.
$$\widetilde{f}(x)=f(\widetilde{x}).$$
- "Răspuns corect la problemă aproape exactă"

- Cele două axiome ale AVF implică stabilitatea regresivă a operației $\circledcirc$
  (1) $\forall x \in \mathbb{R}$, $\exists \epsilon$ cu $|\epsilon| \leq \text{eps}$ a.î. $\text{fl}(x) = x(1+\epsilon)$
  (2) Pentru orice NVF x, y, există ϵ cu $|\epsilon| \leq \text{eps}$ a.î. $x \circledcirc y = (x \circ y)(1+\epsilon)$
- **Exemplu**: Scăderea $f(x_1, x_2) = x_1 - x_2$ cu algoritmul

$$\widetilde{f}(x_1, x_2) = \text{fl}(x_1) \ominus \text{fl}(x_2)$$

- (1) implică existența $|\epsilon_1|, |\epsilon_2| \leq \text{eps}$ a.î.

$$\text{fl}(x_1) = x_1(1+\epsilon_1), \qquad \text{fl}(x_2) = x_2(1+\epsilon_2)$$

(continuarea exemplului)

- (2) implică existența $|\epsilon_3| \leq \text{eps}$ a.î.

$$\mathrm{fl}(x_1) \ominus \mathrm{fl}(x_2) = (\mathrm{fl}(x_1) - \mathrm{fl}(x_2))(1+\epsilon_3)$$

- Combinând, rezultă existența $|\epsilon_4|, |\epsilon_4| \leq 2\text{eps} + O(\text{eps}^2)$ a.î.

$$\begin{aligned}
\mathrm{fl}(x_1) \ominus \mathrm{fl}(x_2) &= (x_1(1+\epsilon_1) - x_2(1+\epsilon_2))(1+\epsilon_3)\\
&= x_1(1+\epsilon_1)(1+\epsilon_3) - x_2(1+\epsilon_2)(1+\epsilon_3)\\
&= x_1(1+\epsilon_4) - x_2(1+\epsilon_5)
\end{aligned}$$

- Deci, $\mathrm{fl}(x_1) - \mathrm{fl}(x_2) = \widetilde{x}_1 - \widetilde{x}_2$

Stabilitatea AVF III

- **Exemplu**: Produsul $f(x, y) = x^* y$ calculat cu $\otimes$ și $\oplus$ este regresiv stabil

Stabilitatea AVF III

- **Exemplu**: Produsul $f(x,y) = x^{*}y$ calculat cu $\otimes$ și $\oplus$ este regresiv stabil
- **Exemplu**: Produsul exterior $f(x,y) = xy^{*}$ calculat cu $\otimes$ nu este regresiv stabil (înafară de cazul când $\widetilde{f}$ are rangul 1)

Stabilitatea AVF III

- **Exemplu**: Produsul $f(x,y)=x^{*}y$ calculat cu $\otimes$ și $\oplus$ este regresiv stabil
- **Exemplu**: Produsul exterior $f(x,y)=xy^{*}$ calculat cu $\otimes$ nu este regresiv stabil (înafară de cazul când $\widetilde{f}$ are rangul 1)
- **Exemplu**: $f(x)=x+1$ calculat cu $\widetilde{f}(x)=\mathrm{fl}(x)\oplus 1$ nu este regresiv stabil (considerăm $x\approx 0$)

Stabilitatea AVF III

- **Exemplu**: Produsul $f(x,y) = x^*y$ calculat cu $\otimes$ și $\oplus$ este regresiv stabil
- **Exemplu**: Produsul exterior $f(x,y) = xy^*$ calculat cu $\otimes$ nu este regresiv stabil (înafară de cazul când $\widetilde{f}$ are rangul 1)
- **Exemplu**: $f(x) = x + 1$ calculat cu $\widetilde{f}(x) = \mathrm{fl}(x) \oplus 1$ nu este regresiv stabil (considerăm $x \approx 0$)
- **Exemplu**: $f(x,y) = x + y$ calculat cu $\widetilde{f}(x,y) = \mathrm{fl}(x) \oplus \mathrm{fl}(y)$ este regresiv stabil

Teoremă (Precizia unui algoritm regresiv stabil)

Daca se utilizează un algoritm regresiv stabil pentru a rezolva problema f cu numărul de condiționare κ, eroarea relativă satisface

$$\frac{\left\|\widetilde{f}(x)-f(x)\right\|}{\|f(x)\|}=O((\operatorname{cond} f)(x)\mathrm{eps})$$

Demonstrație.

Stabilitatea regresivă înseamnă $\widetilde{f}(x)=f(\widetilde{x})$, pentru un anumit $\widetilde{x}$ a. î. $\frac{\|\widetilde{x}-x\|}{\|x\|}=O(\mathrm{eps})$. Definiția numărului de condiționare ne dă

$$\frac{\left\|\widetilde{f}(x)-f(x)\right\|}{\|f(x)\|}=((\operatorname{cond} f)(x)+o(1))\frac{\|\widetilde{x}-x\|}{\|x\|}$$

unde $o(1)\to 0$ la fel ca eps $\to 0$. Combinând aceste două se obține rezultatul dorit. $\square$

Bibliografie I

- James Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- W. Gautschi, *Numerical Analysis. An Introduction*, Birkhäuser, Basel, 1997.
- D. Goldberg, *What every computer scientist should know about floating-point arithmetic*, Computing Surveys **23** (1991), no. 1, 5–48.
- Nicholas J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
- M. L. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*, SIAM, Philadelphia, 2001.
- J. Stoer, R. Burlisch, *Introduction to Numerical Analysis*, 2nd ed., Springer Verlag, 1992.

- C. Überhuber, *Computer-Numerik*, vol. 1, 2, Springer Verlag, Berlin, Heidelberg, New-York, 1995.
- C. Ueberhuber, *Numerical Computation. Methods, Software and Analysis*, vol. I, II, Springer Verlag, Berlin, Heidelberg, New York, 1997.