

Evaluarea algoritmilor: complexitate și erori.

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" București, Facultatea de Inginerie Electrică,
Departamentul de Electrotehnică

Suport didactic pentru disciplina *Algoritmi Numerici*, 2017-2018

Notes

Cuprins

- 1 Complexitatea algoritmilor
 - Timp de calcul
 - Necesar de memorie
- 2 Analiza erorilor (recapitulare)
 - Analiza erorilor de rotunjire
 - Analiza erorilor de trunchiere
 - Analiza erorilor inerente
- 3 Condiționare și stabilitate
 - Condiționare
 - Stabilitate

Notes

$T = O(?)$

Complexitatea unui algoritm din punct de vedere al timpului de calcul

relația dintre timpul de calcul exprimat în număr de operații elementare și dimensiunea problemei

Operație elementară

operația care durează cel mai mult.

Algoritmi polinomiali - de ordinul 1 (liniar)

$p = 0;$

pentru $i = 1, n$

$p = p + a_i b_i$

$T = O(2n) \approx O(n)$

•

Notes

Notes

Algoritmi polinomiali - de ordinul 2 (pătratic)

pentru $i = 1, n$
 $b_i = 0$
 pentru $j = 1, n$
 $b_i = b_i + a_{ij}x_j$ $T = O(2n^2) \approx O(n^2)$

-
-

Algoritmi polinomiali - de ordinul 3 (cubic)

pentru $i = 1, n$
 pentru $j = 1, n$
 $c_{ij} = 0$
 pentru $k = 1, n$
 $c_{ij} = c_{ij} + a_{ik}b_{kj}$

-
-
-

$$T = O(2n^3) \approx O(n^3)$$

Notes

Notes

Algoritmi polinomiali - de ordinul k

$$T = O(n^k) \iff (\exists) C > 0 \text{ și } n_0 \text{ a.i. } T \leq Cn^k \text{ } (\forall) n \geq n_0$$

Algoritm de ordin 0: $T = O(1)$ - nu depinde de dimensiunea problemei.

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < \dots O(e^n) < O(n!)$$

Notes

Ex1: Evaluarea unui polinom - varianta I

$$P(x) = a_0 + a_1x + \dots + a_nx^n. \quad (1)$$

$$P(x) = a_0 + \sum_{i=1}^n a_i x^i \quad (2)$$

; Varianta 1

$$P = a_0;$$

pentru $i = 1, n$

$$t = a_i$$

pentru $j = 1, i$

$$t = t * x$$

• $P = P + t;$

•

Nr. de operații elementare:

$$\sum_{i=1}^n (i + 1) \approx n^2/2$$

$$T_1 = O(n^2/2) \approx O(n^2)$$

Notes

Ex1: Evaluarea unui polinom - varianta II

$$P(x) = a_0 + a_1 t_1 + a_2 t_2 + \dots + a_n t_n = a_0 + \sum_{i=1}^n a_i t_i, \quad (3)$$

unde

$$t_1 = x, \quad t_i = t_{i-1} x, \quad i = 2, \dots, n. \quad (4)$$

; Varianta 2

$$P = a_0;$$

$$t = 1;$$

$$\text{pentru } i = 1, n \quad T_2 = O(3n) \approx O(n)$$

$$t = t * x$$

$$P = P + a_i * t$$

•

Ex1: Evaluarea unui polinom - varianta III

$$P(x) = a_0 + x(a_1 + x(a_2 + \dots x(a_{n-1} + a_n x) \dots)). \quad (5)$$

; Varianta 3

$$P = a_n;$$

$$\text{pentru } i = n - 1, 0, -1 \quad T_3 = O(2n) \approx O(n)$$

$$P = a_i + P * x$$

•

Notes

Notes

Ex2: Reducerea timpului de calcul

; Varianta A

întreg i, j, n

real a, b

...

tablou real $c[n][n]$

pentru $i = 1, n$

pentru $j = 1, n$

$c_{ij} = f(i * a) + f(j * b)$; f definită în altă parte

-
-

Operație elementară: evaluarea funcției $f \Rightarrow T_A = O(2n^2)$.

Notes

Ex2: Reducerea timpului de calcul

; Varianta B

întreg i, j, n

real a, b, p

...

tablou real $c[n][n]$

pentru $i = 1, n$

$p = f(i * a)$

pentru $j = 1, n$

$c_{ij} = p + f(j * b)$

-
-

$T_B = O(n(n + 1)) = O(n^2 + n) \approx O(n^2)$.

Notes

Ex2: Reducerea timpului de calcul

; Varianta C

întreg i, j, n

real a, b

...

tablou real $c[n][n]$

tablou real $p[n], q[n]$

pentru $i = 1, n$

$p_i = f(i * a)$

$q_i = f(i * b)$

$$T_C = O(2n)$$

•

pentru $i = 1, n$

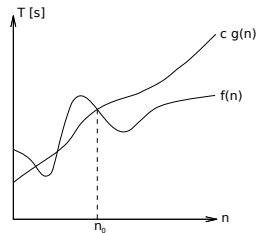
pentru $j = 1, n$

$c_{ij} = p_i + q_j$

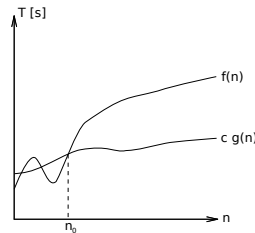
•

•

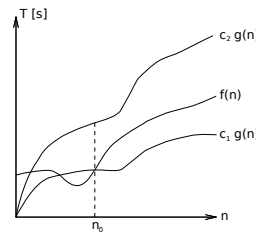
Notății asimptotice



$$T = f(n) = O(g(n))$$



$$T = f(n) = \Omega(g(n))$$



$$T = f(n) = \Theta(g(n))$$

$$T = O(g(n)) \iff \exists C > 0 \text{ și } n_0 \text{ a.i. } T \leq Cg(n) (\forall n \geq n_0)$$

$$T = \Omega(g(n)) \iff \exists C > 0 \text{ și } n_0 \text{ a.i. } Cg(n) \leq T (\forall n \geq n_0)$$

$$T = \Theta(g(n)) \iff \exists C_1 \text{ și } C_2 \text{ și } n_0 \text{ a.i. } C_1g(n) \leq T \leq C_2g(n) (\forall n \geq n_0)$$

Notes

Notes

Exemplu - căutarea unei valori

Fie $x_1 < x_2 < \dots < x_n$ și x_{crt} .

Se cere subintervalul $[x_k, x_{k+1}]$ se află valoarea x_{crt} .

```
funcție caută_v0(n, x, xcrt)
;caută valoarea xcrt în șirul ordonat x cu n valori
întreg n
tablou_real x[n]; șirul este presupus ordonat
real xcrt
...
dacă (xcrt < x1)
    rez = 1
altfel dacă (xcrt > xn)
    rez = n - 1
altfel
    flag = 0
    k = 1
    cât timp ((k <= n - 1) și (flag = 0))
        dacă xcrt ≤ x_{k+1}
            rez = k
            flag = 1
        altfel
            k = k + 1
    întoarce rez
```

Parcurgerea
subintervalelor:

$$T = O(n)$$

$$T = \Omega(1)$$

Notes

Exemplu - căutarea unei valori (căutarea binară)

Fie $x_1 < x_2 < \dots < x_n$ și x_{crt} .

Se cere subintervalul $[x_k, x_{k+1}]$ se află valoarea x_{crt} .

Căutarea
binară:

$T = O(\log(n))$

$$n/2 + n/4 + n/8 +$$

$$\dots + n/2^k = n - 1$$

$$\Rightarrow 2^k = n$$

$$\Rightarrow k = \log_2(n)$$

```
funcție caută(n, x, xcrt)
...
dacă (xcrt < x1)
    rez = 1
altfel dacă (xcrt > xn)
    rez = n - 1
altfel
    k1 = 1
    k2 = n
    cât timp k2 - k1 ≠ 1
        km = [(k1 + k2) / 2] ; [...] este partea întreagă
        dacă xcrt < x_{km}
            k2 = km
        altfel
            k1 = km
    rez = k1
    întoarce rez
```

Notes

Exemplu - căutarea binară, implementare recursivă

```
funcție caută_v2(n, x, xcrt)
...
dacă (xcrt < x1)
    rez = 1
altfel dacă (xcrt > xn)
    rez = n - 1
altfel
rez = binary_search(x, xcrt, 1, n);
•
întoarce rez
;
funcție binary_search(x, xcrt, k1, k2)
;presupuneri:
; x - ordonate, k1 < k2
; xcrt se afla in interiorul vectorului x
...
km = [(k1 + k2)/2]
dacă k2 = k1 + 1
    rez = km
altfel dacă xkm > xcrt
    rez = binary_search(x, xcrt, k1, km)
altfel
    rez = binary_search(x, xcrt, km, k2)
•
întoarce rez
```

M = O(?)

Complexitatea unui algoritm din punct de vedere al necesarului de memorie

relația dintre necesarul de memorie exprimat în număr de locații elementare și dimensiunea problemei

Locația elementară de memorie

locația corespunzătoare unui număr real.

$$M = O(n^k) \iff (\exists) C \text{ a.i. } M \leq Cn^k.$$

Notes

Notes

Ex2: Reducerea timpului de calcul

; Varianta A

întreg i, j, n

real a, b

...

tablou real $c[n][n]$

pentru $i = 1, n$

 pentru $j = 1, n$

$c_{ij} = f(i * a) + f(j * b)$; f definită în altă parte

•
•

$$T_A = O(2n^2)$$

$$M_A = O(n^2 + 2) \approx O(n^2)$$

Notes

Ex2: Reducerea timpului de calcul

; Varianta B

întreg i, j, n

real a, b, p

...

tablou real $c[n][n]$

pentru $i = 1, n$

$p = f(i * a)$

 pentru $j = 1, n$

$c_{ij} = p + f(j * b)$

•
•

$$T_B = O(n(n + 1)) = O(n^2 + n) \approx O(n^2)$$

$$M_B = O(n^2 + 3) \approx O(n^2)$$

Notes

Ex2: Reducerea timpului de calcul

```
; Varianta C
întreg i, j, n
real a, b
...
tablou real c[n][n]
tablou real p[n], q[n]
pentru i = 1, n
    pi = f(i * a)      TC = O(2n)
    qi = f(i * b)      MC = O(n2 + 2n + 2) ≈ O(n2)
    •
    pentru i = 1, n
        pentru j = 1, n
            cij = pi + qj
            •
        •
```

Concluzii (complexitate)

- Resursele de calcul necesare se exprimă cu ajutorul noțiunii de *complexitate*, care se poate referi la două aspecte: timpul de calcul (T) și necesarul de memorie (M).
- Pentru analiza complexității trebuie stabilite una sau mai multe variabile de care depinde dimensiunea problemei.
- Complexitatea T - se exprimă în număr de operații elementare;
- Complexitatea M - se exprimă în număr de locații elementare de memorie.

Notes

Notes

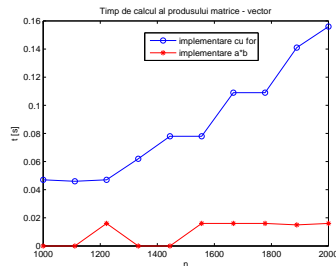
Concluzii (complexitate)

- Cea mai simplă analiză asimptotică a complexității este dată de notația "big O".
- După elaborarea unui algoritm, este necesară analiza complexității lui și investigarea unor modalități de reducere a resurselor de calcul necesare.
- **Elaborarea unui algoritm înseamnă întotdeauna stabilirea unui compromis între timp de calcul și necesar de memorie.**
- **Nu neglijați facilitățile specifice mediului de programare în care lucrați pentru a scrie coduri eficiente (vedeți exemplul următor).**

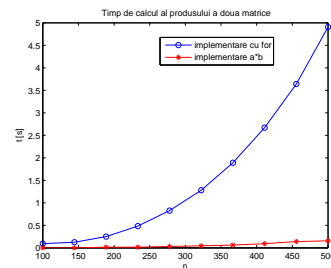
Notes

Implementări eficiente în Matlab

Comparație între două implementări posibile în Matlab.



Timpul de calcul al produsului dintre o matrice pătrată și un vector în funcție de dimensiunea problemei.



Timpul de calcul al produsului dintre două matrice pătrate în funcție de dimensiunea problemei.

Notes

Tipuri de erori

În funcție de tipul cauzelor care le generează:

- 1 **Erori de rotunjire** - datorate reprezentării finite a numerelor reale;
- 2 **Erori de trunchiere** - datorate reprezentării finite a algoritmului;
- 3 **Erori inerente** - datorate reprezentării imprecise a datelor de intrare.

Notes

Mărimi utile - eroarea absolută și marginea ei

Fie:

$\mathbf{x} \in \mathbb{R}^n$ - valoarea exactă a unei mărimi;

$\bar{\mathbf{x}}$ - valoarea aproximativă.

Eroarea absolută $\mathbf{e}_x \in \mathbb{R}^n$:

$$\mathbf{e}_x = \bar{\mathbf{x}} - \mathbf{x}. \quad (6)$$

Marginea erorii absolute $a_x \in \mathbb{R}$:

$$\|\mathbf{e}_x\| \leq a_x. \quad (7)$$

Dacă $n = 1$ rezultă

$$\bar{x} - a_x \leq x \leq \bar{x} + a_x. \quad (8)$$

Echivalentă cu: $x \in [\bar{x} - a_x, \bar{x} + a_x]$.

Scrisă pe scurt ca:

$$"x = \bar{x} \pm a_x". \quad (9)$$

Notes

Mărimi utile - eroarea relativă și marginea ei

Eroarea relativă $\varepsilon_{\mathbf{x}} \in \mathbb{R}^n$:

$$\varepsilon_{\mathbf{x}} = \frac{\mathbf{e}_{\mathbf{x}}}{\|\mathbf{x}\|}. \quad (10)$$

Marginea erorii relative $r_x \in \mathbb{R}$

$$\|\varepsilon_{\mathbf{x}}\| \leq r_x. \quad (11)$$

Cel mai adesea, r_x se exprimă în procente.
 Scriere pe scurt:

$$" \mathbf{x} = \bar{\mathbf{x}} \pm r_x \% ". \quad (12)$$

Notes

Exemplu: π

$$x = 3.1415\dots$$

$$\bar{x} = 3.14$$

$$e_x = -0.0015\dots$$

$$a_x = 0.0016$$

$$\varepsilon_x = -0.0015\dots / 3.1415\dots$$

$$r_x = 0.0016/3 \leq 0.0006 = 0.06\%.$$

$$\pi = 3.14 \pm 0.0016 \quad \text{sau} \quad \pi = 3.14 \pm 0.06\%.$$

Notes

Concluzii

Relația " $\mathbf{x} = \bar{\mathbf{x}} \pm a_x$ "

unde $\mathbf{x}, \bar{\mathbf{x}} \in \mathbb{R}^n$ și $a_x \in \mathbb{R}$ se interpretează astfel:

$$(\exists) \mathbf{e}_x \in \mathbb{R}^n, \|\mathbf{e}_x\| \leq a_x, \text{ astfel încât } \bar{\mathbf{x}} = \mathbf{x} + \mathbf{e}_x, \quad (13)$$

Relația " $\mathbf{x} = \bar{\mathbf{x}} \pm r_x \%$ "

unde $\mathbf{x}, \bar{\mathbf{x}} \in \mathbb{R}^n$, $r_x \% = 100r_x$ și $r_x \in \mathbb{R}$ se interpretează astfel:

$$(\exists) \varepsilon_x \in \mathbb{R}^n, \|\varepsilon_x\| \leq r_x, \text{ astfel încât } \bar{\mathbf{x}} = \mathbf{x} + \|\mathbf{x}\| \varepsilon_x. \quad (14)$$

În cazul unei mărimi scalare ($n = 1$), relația (14) se scrie

$$\bar{x} = x(1 \pm \varepsilon_x), \quad (15)$$

semnul plus corespunzând unei valori x pozitive, iar semnul minus uneia negative.

Notes

Cifre semnificative

Reprezentarea unui număr real în baza 10:

$$\bar{x} = f \cdot 10^n. \quad (16)$$

unde $0.1 \leq |f| < 1$.

Cifrele părții fracționare se numesc **cifre semnificative**.

Exemple:

$$3.14 = 0.314 \cdot 10^1$$

$$-0.007856 = -0.7856 \cdot 10^{-2}.$$

Notes

Rotunjirea afectează reprezentarea numerelor reale

$$\bar{x} = 0.\overbrace{***\dots*}^f \cdot 10^n, \quad (17)$$

k cifre

$$x = 0.\overbrace{***\dots*} \#\#\# \dots \cdot 10^n, \quad (18)$$

k cifre

$$e_x = \bar{x} - x = -0.\underbrace{000\dots0}_k \#\#\# \dots \cdot 10^n = -0.\#\#\# \dots \cdot 10^{n-k}, \quad (19)$$

Rotunjirea afectează reprezentarea numerelor reale

$$\varepsilon_x = \frac{e_x}{x} = \frac{-0.\#\#\# \dots \cdot 10^{n-k}}{0.\overbrace{***\dots*} \#\#\# \dots \cdot 10^n} = -\frac{0.\#\#\# \dots}{0.*** \dots} 10^{-k} \quad (20)$$

k cifre

$$|\varepsilon_x| \leq \frac{1}{0.1} 10^{-k} = 10^{-k+1}. \quad (21)$$

Marginea erorii relative de rotunjire a unui sistem de calcul depinde doar de numărul de cifre semnificative ce pot fi memorate. Pentru un sistem de calcul ce lucrează cu k cifre semnificative, marginea erorii relative de rotunjire este 10^{-k+1} .

Notes

Notes

Rotunjirea afectează calculele

Adunarea a două numere reale

Intuitiv: pp. $k = 3$, $x_1 + x_2 = ?$

$$x_1 = 3.73 = 0.373 \cdot 10^1$$

$$x_2 = 0.006 = 6 \cdot 10^{-3}$$

$$x_2 = 6 \cdot 10^{-4} \cdot 10^1 = 0.0006 \cdot 10^1 = 0.000 \cdot 10^1$$

Rezultat: $x_1 + x_2 = x_1$.

Zeroul mașinii

Zeroul (acuratețea, precizia, "epsilon-ul") mașinii = cel mai mic ϵ pentru care $1 + \epsilon > 1$.

- $(\forall) a < \epsilon$, $1 + a = 1$ (în calculator)
- în mod uzual $\epsilon = 2.22 \cdot 10^{-16}$.
- Matlab: `eps`
- Scilab `%eps`.
- Zeroul mașinii nu trebuie confundat cu cel mai mic număr reprezentabil în calculator și care, în mod uzual are valoarea $2.23 \cdot 10^{-308}$.

Consecință: adunarea numerelor reale în calculator nu este asociativă.

Notes

Notes

Determinarea eps într-un mediu de programare

```
funcție zeroul_mașinii ()  
real eps  
eps = 1  
cât timp (1 + eps > 1)  
    eps = eps/2  
•  
eps = eps*2  
întoarce eps
```

Notes

Notații pentru un număr real

- **Notație științifică:** $x = a \cdot 10^b$ unde $0.1 \leq |a| < 1$
- **Notația științifică normalizată:** $x = a \cdot 10^b$ unde $1 \leq |a| < 10$ (pune în evidență ordinul de mărime)
- **Notația inginerescă** în care b se alege numai dintre multiplii lui 3, și, deci a poate avea valori între $1 \leq a < 1000$. (se citește cu prefixe: mega, kilo, mili, micro etc.)

Reprezentarea numerelor reale în calculator este un fel de echivalent hardware al notației științifice.

Notes

Reprezentarea unui număr (standardul IEEE)

- Reprezentarea intervalului [1,2]:
 $\{1, 1 + 2^{-52}, 1 + 2 \cdot 2^{-52}, 1 + 3 \cdot 2^{-52}, \dots, 2\}$.
 Spațiile dintre numerele vecine = zeroul mașinii.
- Reprezentarea intervalului $[2^j, 2^{j+1}]$: mulțimea de mai sus înmulțită cu 2^j . Spațiile dintre numerele vecine sunt scalate în funcție de dimensiunea numerelor.

Zeroul mașinii reflectă rezoluția mulțimii reale discrete \mathcal{R} ce poate fi reprezentată în calculator. Proprietate:

$$(\forall)x \in \mathbb{R}, (\exists)\bar{x} \in \mathcal{R} \text{ astfel încât } |x - \bar{x}| \leq \text{eps}|x|. \quad (23)$$

Eroarea relativă dintre numărul real x și reprezentarea lui discretă \bar{x} este mărginită superior de zeroul mașinii.

$(\exists)\varepsilon$, unde $|\varepsilon| \leq \text{eps}$ a.i:

$$\bar{x} = x(1 + \varepsilon). \quad (24)$$

 39/72

Notes

Exemplu

$$f(x) = f(x_0) + \frac{x - x_0}{1!} f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \dots \quad (25)$$

sinus, $x_0 = 0$:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}. \quad (26)$$

$$\bar{s} = \sum_{k=0}^n (-1)^k \frac{x^{2k+1}}{(2k+1)!}. \quad (27)$$

$$|e_s| = |\bar{s} - s| \leq \frac{x^{2n+1}}{(2n+1)!}. \quad (28)$$

 40/72

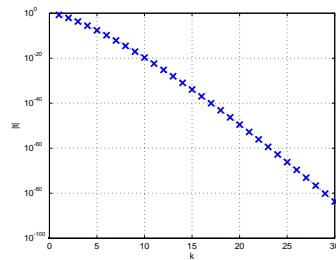
Notes

Algoritm cu controlul erorii de trunchiere

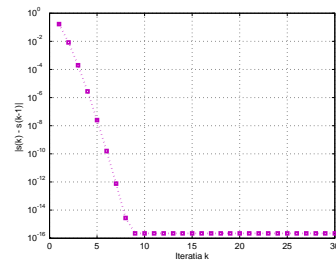
```
funcție sinus(x, e)
; întoarce valoarea funcției sinus în punctul x
; prin trunchierea seriei Taylor dezvoltată în 0
real x           ; punctul în care se va evalua funcția sin
real e           ; eroarea de trunchiere impusă
real t, s
întreg k
t = x
s = t
k = 0
cât timp (|t| > e)
    k = k + 1
    t = (-1) * t * (x^2 / ((2k)(2k+1)))
    s = s + t
•
întoarce s
```

Notes

Rezultate numerice



Modulul termenului curent al dezvoltării în serie Taylor a funcției sinus.



Modulul diferenței dintre sume parțiale consecutive la dezvoltarea în serie Taylor a funcției sinus.

Notes

Efectul perturbațiilor datelor de intrare

$$y = f(x_1, x_2, \dots, x_n). \quad (29)$$

$$dy = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + \dots + \frac{\partial f}{\partial x_n} dx_n. \quad (30)$$

$$\Delta y \approx \frac{\partial f}{\partial x_1} \Delta x_1 + \frac{\partial f}{\partial x_2} \Delta x_2 + \dots + \frac{\partial f}{\partial x_n} \Delta x_n. \quad (31)$$

$$\Delta x_k = \bar{x}_k - x_k = e_{x_k}, \quad (32)$$

Notes

Eroarea absolută a rezultatului și marginea ei

$$e_y = \bar{y} - y = \Delta y:$$

$$e_y = \sum_{k=1}^n \frac{\partial f}{\partial x_k} e_{x_k}. \quad (33)$$

$$\left| \sum_{k=1}^n \frac{\partial f}{\partial x_k} e_{x_k} \right| \leq \sum_{k=1}^n \left| \frac{\partial f}{\partial x_k} e_{x_k} \right| = \sum_{k=1}^n \left| \frac{\partial f}{\partial x_k} \right| |e_{x_k}| \leq \sum_{k=1}^n \left| \frac{\partial f}{\partial x_k} \right| a_{x_k}, \quad (34)$$

unde $|e_{x_k}| \leq a_{x_k}$.

Marginea erorii absolute a rezultatului

$$a_y = \sum_{k=1}^n \left| \frac{\partial f}{\partial x_k} \right| a_{x_k}. \quad (35)$$

Notes

Eroarea relativă a rezultatului și marginea ei

$$\varepsilon_y = \mathbf{e}_y/|y|$$

$$\varepsilon_y = \frac{\sum_{k=1}^n \frac{\partial f}{\partial x_k} \mathbf{e}_{x_k}}{|y|} = \sum_{k=1}^n \frac{\partial f}{\partial x_k} \frac{\mathbf{e}_{x_k}}{|y|} = \sum_{k=1}^n \frac{\partial f}{\partial x_k} \frac{|x_k|}{|y|} \varepsilon_{x_k}. \quad (36)$$

Marginea erorii relative a rezultatului

$$r_y = \sum_{k=1}^n \left| \frac{\partial(\ln f)}{\partial x_k} \right| |x_k| r_{x_k}. \quad (37)$$

Notes

Cazuri particulare: +, -

Erori	Adunare $y = x_1 + x_2$	Scădere $y = x_1 - x_2$
Eroare absolută: $\mathbf{e}_y =$	$\mathbf{e}_{x_1} + \mathbf{e}_{x_2}$	$\mathbf{e}_{x_1} - \mathbf{e}_{x_2}$
majorată de: $a_y =$	$a_{x_1} + a_{x_2}$	$a_{x_1} + a_{x_2}$
Eroare relativă: $\varepsilon_y =$	$\frac{x_1}{x_1+x_2} \varepsilon_{x_1} + \frac{x_2}{x_1+x_2} \varepsilon_{x_2}$	$\frac{x_1}{x_1-x_2} \varepsilon_{x_1} - \frac{x_2}{x_1-x_2} \varepsilon_{x_2}$
majorată de $r_y =$	$\frac{x_1}{x_1+x_2} r_{x_1} + \frac{x_2}{x_1+x_2} r_{x_2}$	$\frac{x_1}{x_1-x_2} r_{x_1} + \frac{x_2}{x_1-x_2} r_{x_2}$

Erorile rezultatului adunării și scăderii a două numere reale în funcție de erorile datelor de intrare.

NB! La adunare și scădere marginile erorilor absolute se adună.

- Adunarea este o operație bine condiționată.
- Scăderea este o operație prost condiționată.

Notes

Exemplu

$$x_1 = 1.23 \pm 1\% , x_2 = 1.22 \pm 1\%$$

- Scădere:

$$r = |1.23/0.01 \cdot 1/100 + 1.22/0.01 \cdot 1/100 = 1.23 + 1.22 = 2.45 = 245\%$$

$$x_1 - x_2 = 0.01 \pm 245\%.$$

- Adunare:

$$r = |1.23/2.45 \cdot 1/100 + 1.22/2.45 \cdot 1/100 \approx 0.5 \cdot 1/100 + 0.5 \cdot 1/100 = 1/100 = 1\%.$$

$$x_1 + x_2 = 2.45 \pm 1\%.$$

Notes

Cazuri particulare: *, /

Erori	Înmulțire $y = x_1 x_2$	Împărțire $y = \frac{x_1}{x_2}$
Eroare absolută: $e_y =$	$x_2 e_{x_1} + x_1 e_{x_2}$	$\frac{1}{x_2} e_{x_1} - \frac{x_1}{x_2^2} e_{x_2}$
majorată de: $a_y =$	$ x_2 a_{x_1} + x_1 a_{x_2}$	$\frac{1}{ x_2 } a_{x_1} + \frac{ x_1 }{x_2^2} a_{x_2}$
Eroare relativă: $e_y =$	$e_{x_1} + e_{x_2}$	$e_{x_1} - e_{x_2}$
majorată de: $r_y =$	$r_{x_1} + r_{x_2}$	$r_{x_1} + r_{x_2}$

Erorile rezultatului înmulțirii și împărțirii a două numere reale în funcție de erorile datelor de intrare.

NB! La înmulțire și împărțire marginile erorilor relative se adună.

- Înmulțirea și împărțirea sunt operații bine condiționate.

Notes

Scăderea trebuie evitată

$$ax^2 + bx + c = 0$$
$$x_{1,2} = (-b \pm \sqrt{b^2 - 4ac}) / (2a)$$

pp. $b > 0$ și că $b^2 \gg 4ac$

dacă $b > 0$

$$x_1 = (-b - \sqrt{b^2 - 4ac}) / (2a)$$

altfel

$$x_1 = (-b + \sqrt{b^2 - 4ac}) / (2a)$$

- $x_2 = c / (a * x_1)$

Extragerea radicalului

$$y = \sqrt{x}$$

$$e_y = \frac{df}{dx} e_x = \frac{1}{2\sqrt{x}} e_x, \quad (38)$$

$$\varepsilon_y = \frac{e_y}{y} = \frac{1}{2\sqrt{x}\sqrt{x}} e_x = \frac{e_x}{2x} = \frac{\varepsilon_x}{2}. \quad (39)$$

Dar rotunjirea nu poate fi ignorata!

Notes

Notes

Superpoziția erorilor

eroarea relativă într-un calcul aproximativ
=
eroarea relativă produsă de calculul aproximativ cu numere exacte (eroarea de rotunjire)
+
eroarea relativă produsă de calculul exact cu numere aproximative (afectate deci de erori inerente).

$\bar{y} = y_i(1 + \text{eps}) = y(1 + \varepsilon_y)(1 + \text{eps}) \approx y(1 + \varepsilon_y + \text{eps})$,
de unde $(\bar{y} - y)/y = \varepsilon_y + \text{eps}$.

$$\varepsilon_{\sqrt{x}} = \frac{\varepsilon_x}{2} + \text{eps}. \quad (40)$$

Eroarea relativă a oricărui rezultat numeric este cel puțin egală cu zeroul mașinii.

Condiționare vs. stabilitate

Condiționarea

se referă la comportarea **problemei matematice** la perturbații ale datelor.

Stabilitatea

se referă la comportarea **algoritmului** la perturbații ale datelor.

Notes

Notes

Condiționare

Problemă matematică f formulată explicit:

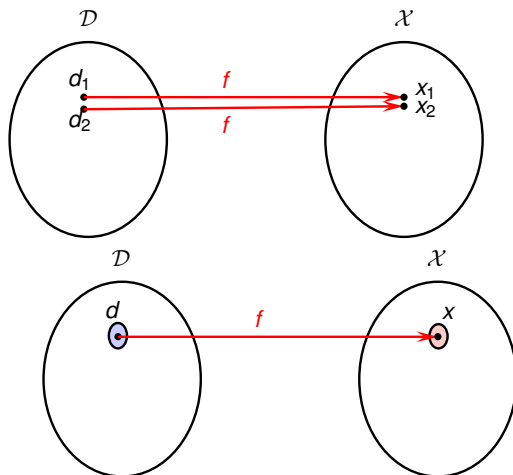
Fie $f : \mathcal{D} \rightarrow \mathcal{X}$ și $\mathbf{d} \in \mathcal{D}$.

Să se găsească $\mathbf{x} \in \mathcal{X}$ astfel încât $f(\mathbf{d}) = \mathbf{x}$. (41)

O problemă este bine condiționată dacă perturbații mici ale datelor conduc la perturbații mici ale rezultatului.

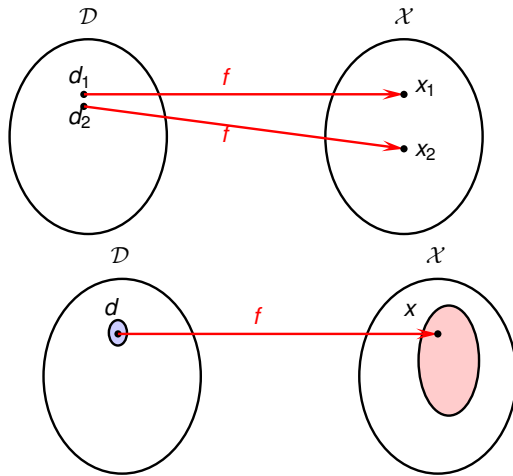
Notes

Reprezentări intuitive - problemă bine condiționată



Notes

Reprezentări intuitive - problemă prost condiționată



Condiționare

Problemă matematică poate fi formulată și implicit:

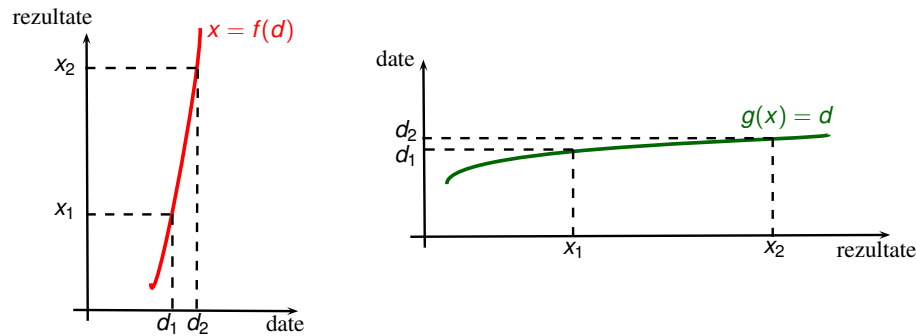
Fie $g : \mathcal{X} \rightarrow \mathcal{D}$ și $\mathbf{d} \in \mathcal{D}$.

Să se găsească $\mathbf{x} \in \mathcal{X}$ astfel încât $g(\mathbf{x}) = \mathbf{d}$. (42)

Notes

Notes

Reprezentări intuitive - problemă prost condiționată



Numărul de condiționare absolut

$$\hat{\kappa} = \lim_{\delta \rightarrow 0} \sup_{\|\delta \mathbf{d}\| < \delta} \frac{\|\delta f\|}{\|\delta \mathbf{d}\|}, \quad (43)$$

sau, într-o scriere simplificată

$$\hat{\kappa} = \sup_{\|\delta \mathbf{d}\|} \frac{\|\delta f\|}{\|\delta \mathbf{d}\|}, \quad (44)$$

unde $\delta f = f(\mathbf{d} + \delta \mathbf{d}) - f(\mathbf{d})$ și $\delta \mathbf{d}$ sunt mărimi infinitesimale.

Dacă f este derivabilă, atunci perturbația δf se poate aproxima în condiția $\|\delta \mathbf{d}\| \rightarrow 0$ ca

$$\delta f = \mathbf{J}(\mathbf{d})\delta \mathbf{d}, \quad (45)$$

$$\hat{\kappa} = \|\mathbf{J}(\mathbf{d})\|. \quad (46)$$

Notes

Notes

Numărul de condiționare relativ

$$\kappa = \lim_{\delta \rightarrow 0} \sup_{\|\delta \mathbf{d}\| < \delta} \frac{\|\delta f\| / \|f(\mathbf{d})\|}{\|\delta \mathbf{d}\| / \|\mathbf{d}\|}, \quad (47)$$

sau, scris mai simplu în ipoteza unor variații infinitesimale

$$\kappa = \sup_{\|\delta \mathbf{d}\|} \frac{\|\delta f\| / \|f(\mathbf{d})\|}{\|\delta \mathbf{d}\| / \|\mathbf{d}\|}. \quad (48)$$

Dacă f este derivabilă, atunci

$$\kappa = \frac{\|\mathbf{J}(\mathbf{d})\|}{\|f(\mathbf{d})\| / \|\mathbf{d}\|}. \quad (49)$$

O problemă este bine condiționată dacă valoarea lui κ este mică și prost condiționată dacă valoarea lui κ este mare. Ce înseamnă mic sau mare, depinde de problemă.

59/72

Exemplu

Numărul de condiționare al scăderii a două numere reale

$f(\mathbf{d}) = d_1 - d_2$, unde $\mathbf{d} = [d_1, d_2]^T$.

$\mathbf{J} = [\partial f / \partial d_1, \partial f / \partial d_2]^T = [1, -1]^T$

$$\kappa = \frac{\|\mathbf{J}(\mathbf{d})\|}{\|f(\mathbf{d})\| / \|\mathbf{d}\|} = \frac{1}{|d_1 - d_2| / \max\{|d_1|, |d_2|\}}. \quad (50)$$

Scăderea este prost condiționată dacă $d_1 \approx d_2$.

60/72

Notes

Notes

Relația între numărul de condiționare, eroare și reziduu

Deoarece

$$\kappa = \sup_{\|\delta \mathbf{d}\|} \frac{\|\delta f\| / \|f(\mathbf{d})\|}{\|\delta \mathbf{d}\| / \|\mathbf{d}\|}. \quad (51)$$

rezultă

$$\frac{\|\delta f\| / \|f\|}{\|\delta \mathbf{d}\| / \|\mathbf{d}\|} \leq \kappa. \quad (52)$$

Perturbația rezultatului este o distanța în spațiul soluțiilor \mathcal{X} , deci reprezintă o eroare absolută $\mathbf{e}_x = \delta f$ sau relativă

$$\varepsilon_x = \delta f / \|f\|.$$

Perturbația datelor, numită și **reziduu** este o distanță în spațiul \mathcal{D} . Reziduu poate fi absolut $\mathbf{e}_d = \delta \mathbf{d}$, unde $\delta \mathbf{d} = \tilde{\mathbf{d}} - \mathbf{d}$, sau relativ $\varepsilon_d = \delta \mathbf{d} / \|\mathbf{d}\|$.

Cu aceste notații:

Notes

Condiționare - concluzie

$$\|\mathbf{e}_x\| \leq \kappa \|\varepsilon_d\|. \quad (54)$$

- Eroarea și reziduu sunt legate prin numărul de condiționare.
- Pentru o problemă cu număr de condiționare mic, o perturbație mică în date va duce la o perturbație mică a rezultatului.
- Problemele matematice care au κ mare sunt prost condiționate și ele nu pot fi rezolvate cu ajutorul calculatorului. Pentru astfel de probleme, trebuie găsită o formulare matematică echivalentă din punct de vedere al rezultatului, dar bine condiționată.

Notes

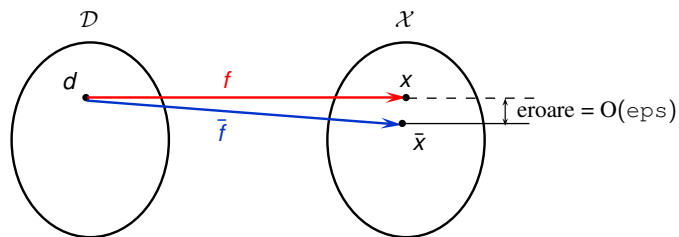
Notes

În cele ce urmează vom presupune că problema f este bine condiționată și pentru rezolvarea ei a fost conceput un algoritm \bar{f} .

Acuratețea unui algoritm

Notes

Acuratețea unui algoritm se referă la eroarea soluției numerice.



Reprezentarea intuitivă a unui algoritm a cărui precizie este ideală.

În mod ideal, un algoritm este precis dacă:

$$\frac{\|\bar{f}(\mathbf{d}) - f(\mathbf{d})\|}{\|f(\mathbf{d})\|} = O(\text{eps}). \quad (55)$$

$\bar{f}(\mathbf{d})$ = "rezultatul algoritmului \bar{f} aplicat datelor \mathbf{d} ".

Stabilitatea unui algoritm

Dar, rotunjirea datelor este inevitabilă, erorile se acumulează și perturbă rezultatul. Este mai util să se țintească **stabilitatea algoritmului**.

Stabilitatea unui algoritm se referă la comportarea algoritmului atunci când datele de intrare sunt perturbate.

Un algoritm \bar{f} folosit pentru rezolvarea unei probleme f este stabil dacă

$$\frac{\|\bar{f}(\bar{\mathbf{d}}) - f(\mathbf{d})\|}{\|f(\mathbf{d})\|} = O(\epsilon_{\text{ps}}), \quad (56)$$

pentru $(\forall)\bar{\mathbf{d}}, \mathbf{d}$ care satisfac $\|\bar{\mathbf{d}} - \mathbf{d}\|/\|\mathbf{d}\| = O(\epsilon_{\text{ps}})$.

Pe scurt, **un algoritm stabil dă răspunsul aproape corect pentru date reprezentate aproape precis.**

Notes

Ilustrarea stabilității unui algoritm - problema

$$\mathbf{Ax} = \mathbf{b}, \quad \text{unde } \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

$$\begin{aligned} x_2 &= 1 \\ x_1 + x_2 &= 0 \end{aligned} \quad (57)$$

$x_1 = -1, x_2 = 1. \mathbf{x} = f(\mathbf{d}) = [-1, 1]^T$.

Să considerăm acum că datele au fost perturbate:

$$\bar{\mathbf{A}} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix},$$

$$\begin{aligned} 10^{-20}x_1 + x_2 &= 1 \\ x_1 + x_2 &= 0 \end{aligned} \quad (58)$$

$x'_1 = -x'_2 = 1/(10^{-20} - 1) \approx -1$. Se poate demonstra că această problemă este bine condiționată.

Notes

Ilustrarea stabilității unui algoritm - algoritmul \bar{f}_1

- **Pasul 1:** se înmulțește prima ecuație a sistemului cu (-10^{20}) și se adună cu a doua, rezultând x_2 ;
- **Pasul 2:** se calculează x_1 din prima ecuație.

La pasul 1 se ajunge la ecuația $(1 - 10^{20})x_2 = -10^{20}$ care, în calculator devine datorită rotunjirilor $-10^{20}x_2 = -10^{20}$, de unde va rezulta $x_2 = 1$, ceea ce este corect.

La pasul 2 ecuația de rezolvat devine $10^{-20}x_1 + 1 = 1$, de unde va rezulta $x_1 = 0$, ceea ce este greșit, foarte departe de valoarea adevărată.

Acest algoritm este instabil.

Ilustrarea stabilității unui algoritm - algoritmul \bar{f}_2

- **Pasul 1:** se înmulțește a doua ecuație a sistemului cu (-10^{-20}) și se adună cu prima, rezultând x_2 ;
- **Pasul 2:** se calculează x_1 din a doua ecuație.

La pasul 1 se ajunge la ecuația $(1 - 10^{-20})x_2 = 1$, care în calculator devine $x_2 = 1$.

La pasul 2 ecuația de rezolvat este $x_1 + 1 = 0$, de unde $x_1 = -1$, ceea ce este corect.

Algoritmul \bar{f}_2 este stabil. Stabilitatea lui este foarte puternică, el a dat raspunsul exact pentru date de intrare aproape precise.

Notes

Notes

Concluzii - estimarea acurateții unei soluții numerice

- 1 Se estimează numărul de condiționare al problemei. Se continuă numai dacă problema matematică este bine condiționată.
- 2 Se investighează stabilitatea algoritmului. Cel mai simplu este ca acest lucru să se realizeze experimental, rulându-se algoritmul pentru date perturbate. Dacă dispersia rezultatelor este mare atunci algoritmul este instabil și trebuie schimbat.
- 3 Dacă algoritmul este stabil, atunci acuratețea finală (modulul erorii relative) este majorată de produsul dintre numărul de condiționare și modulul reziduuului relativ.

Despre un algoritm stabil care generează erori mici pentru probleme bine condiționate se spune că este **robust**.

Notes

Referințe (pseudocod, complexitate)

- [Ciuprina13a] Gabriela Ciuprina - Algoritmi numerici pentru calcule științifice în ingineria electrică, Editura MatrixROM, 2013, paragrafele 1.1. și 1.2.

disponibilă la http://www.lmn.pub.ro/~gabriela/books/AlgNr_MatrixRom2013.pdf

- [Cormen09] T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Introduction to Algorithms*, MIT Press, cap 1 (The role of algorithms in computing), cap 2 (Getting started), cap 3 (Growth of functions).

ediția mai veche este disponibilă online [aici](#)

Notes

Referințe (Matlab)

- [MatlabDoc] Documentația Matlab
disponibilă online la
<http://www.mathworks.com/access/helpdesk/help/helpdesk.html>
- [Moler04] Clever Moler - Numerical Computing with Matlab, SIAM, 2004
disponibilă online la <http://www.mathworks.com/moler/>
- [Getreuer09] Pascal Getreuer - Writing fast Matlab code, 2009
disponibilă online la
<https://www.mathworks.com/matlabcentral/fileexchange/5685-writing-fast-matlab-code>

Referințe (erori, condiționare, stabilitate)

- [Ciuprina13a] Gabriela Ciuprina - Algoritmi numerici pentru calcule științifice în ingineria electrică , Editura MatrixROM, 2013, paragraful 1.3.
disponibilă la http://www.lmn.pub.ro/~gabriela/books/AlgNr_MatrixRom2013.pdf
- [Trefethen97] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM. 1997. Philadelphia, PA. (Lecture 12: Conditioning and Condition Numbers; Lecture 13: Floating Point Arithmetic; Lecture 14: Stability; Lecture 15: More on Stability;)
Cartea v-o pot împrumuta la cerere.
Vă recomand cu căldură să vizitați pagina Prof. Trefethen disponibilă la
<https://people.maths.ox.ac.uk/trefethen/>

Notes

Notes
