Laboratorul de **M**etode **N**umerice

Universitatea "POLITEHNICA" Bucureşti, Facultatea de Electrotehnică, http://lmn.pub.ro, lmn@lmn.pub.ro

Laborator - metode numerice în ingineria electrică

L2 - Erori în rezolvarea problemelor numerice

Autori:

Conf. Gabriela Ciuprina
Prof. Daniel Ioan
Prep. Marius Piper
As. Marius Rădulescu
As. Mihai Popescu

12 martie 2004

Cuprins

Capitolul 2

Erori în rezolvarea problemelor numerice

2.1 Caracterizarea lucrării

În majoritatea cazurilor, algoritmii cu caracter numeric, după implementarea lor pe un sistem de calcul determină nu soluţia exactă a problemei ci o aproximare numerică a acesteia.

Scopul acestei lucrări este de a evidenţia modul în care pot fi caracterizate erorile numerice, motivele apariţiei acestora şi modul în care acestea se propagă. Se studiază *erorile inerente* (datorate datelor de intrare), *erorile de rotunjire* (datorate reprezentării finite, aproximative a numerelor reale) şi *erorile de trunchiere* (datorate aproximării finite a unor procese teoretic infinite).

2.2 Principiul lucrării

Pentru a caracteriza abaterea unei variabile numerice de la valoarea sa exactă se poate folosi *eroarea absolută*, definită prin:

$$e_x = x - x^*, \tag{2.1}$$

în care x este valoarea exactă a variabilei, iar x^* este valoarea sa aproximativă. Deoarece în majoritatea cazurilor valoarea exactă nu este cunoscută se preferă utilizarea unei *margini superioare a erorii absolute*:

$$a_x \geq |e_x| = |x - x^*|. \tag{2.2}$$

În acest caz, în locul valorii exacte x se operează cu intervalul:

$$x^* - a_x \leq x \leq x^* + a_x. \tag{2.3}$$

O altă modalitate de caracterizare a abaterii unei variabile de la valoarea exactă este *eroarea relativă*:

$$\varepsilon_x = \frac{e_x}{x} = \frac{x - x^*}{x}, \tag{2.4}$$

sau *marginea superioară a acesteia*:

$$\Gamma_x = \frac{a_x}{|x|} \geq |eps_x| = \frac{|x - x^*|}{|x|}. \tag{2.5}$$

În majoritatea calculelor tehnice, mărimile rezultate în urma măsurării sunt cunoscute cu 3 maxim 6 cifre semnificative exacte, ceea ce corespunde unor erori relative cuprinse în intervalul $10^{-2} \div 10^{-5}$.

2.2.1 Erori de rotunjire

Una din cauzele frecvente de eroare în calculele cu numere reale se datorează reprezentării finite a acestor numere în sistemele de calcul.

Modul uzual de reprezentare a unui număr real într-un sistem de calcul este de forma:

$$x^* = \pm 0, kkk \ldots k \cdot 10^{\pm kk} = m \cdot 10^e, \tag{2.6}$$

unde k este cifră $(0, 1, \ldots, 9)$, m se numeşte mantisa, iar e exponent. Cu excepţia numărului nul, la care $m = 0$, în rest mantisa satisface relaţia:

$$0.1 \leq |m| \leq 1. \tag{2.7}$$

Presupunând că mantisa este reprezentată cu n cifre, rezultă că prin această reprezentare cifrele "l" din reprezentarea exactă:

$$x = \pm 0. \underbrace{kk \ldots k}_{n} ll \ldots \cdot 10^{\pm kk} \tag{2.8}$$

sunt pierdute prin rotunjire.

În consecinţă, eroarea relativă de rotunjire

$$\varepsilon_x = \frac{|x - x^*|}{|x|} = \frac{0, 0 \ldots 0lll \ldots \cdot 10^{\pm kk}}{0, kk \ldots ll \ldots \cdot 10^{\pm kk}} = \frac{0, lll \ldots}{0.kk \ldots} 10^{-n} = 10^{-n+1} \tag{2.9}$$

depinde de numărul de cifre semnificative folosite în reprezentarea numărului real şi nu de valoarea numărului. Această eroare relativă de rotunjire, specifică sistemului de calcul (calculator + mediu de programare) este cel mai mare număr real, care adăugat la unitate nu "modifică" valoarea acesteia. Ordinul de mărime al erorii relative de rotunjire în sistemele uzuale de calcul este $10^{-5} \div 10^{-20}$ şi poate fi determinat pe fiecare sistem de calcul cu următorul algoritm:

```
; calculează eroarea relativă de rotunjire
real err
err=1
repetă
        err=err/2
până când (1 + err = 1)
scrie err
```

Eroarea relativă de rotunjire *err* este cunoscută sub numele de *zeroul maşinii* şi nu trebuie confundată cu cel mai mic număr pozitiv, nenul, reprezentabil în calculator.

2.2.2 Erori inerente

Datele de intrare folosite în rezolvarea unei probleme tehnice provin în multe cazuri din determinări experimentale. Rezultatul oricărei măsurători este susceptibil de erori şi chiar dacă datele de intrare sunt cunoscute cu maximă precizie ele pot fi afectate de erori de rotunjire.

Erorile datelor de intrare într-un algoritm se numesc erori inerente. Aceste erori se propagă în procesul de calcul şi afectează în final soluţia problemei. Chiar dacă algoritmul de calcul nu poate fi făcut responsabil de prezenţa erorilor în datele de intrare, el poate influenţa precizia soluţiei. Un algoritm la care eroarea relativă a soluţiei nu depăşeşte erorile relative ale datelor de intrare este un *algoritm stabil din punct de vedere numeric.* În schimb, dacă erorile relative ale soluţiei sunt mult mai mari decât cele ale datelor de intrare se spune că algoritmul de rezolvare prezintă instabilităţi numerice. În acest caz este posibil ca abateri foarte mici ale datelor de intrare să determine abateri mari ale soluţiei numerice faţă de cea exactă, şi să facă soluţia numerică inutilizabilă. Dacă se noteaza cu y soluţia problemei, iar cu $x_1, x_2, ..., x_n$ datele acesteia, procedeul de calcul va consta în fond în evaluarea funcţiei $y = f(x_1, x_2, ..., x_n)$

Considerând erorile absolute suficient de mici, acestea se pot aproxima cu diferenţiala:

$$\begin{aligned} \mathrm{d}y &= \frac{\partial f}{\partial x_1}\mathrm{d}x_1 + \frac{\partial f}{\partial x_2}\mathrm{d}x_2 + \cdots + \frac{\partial f}{\partial x_n}\mathrm{d}x_n, \\ e_y &= \frac{\partial f}{\partial x_1}e_{x_1} + \frac{\partial f}{\partial x_2}e_{x_2} + \cdots + \frac{\partial f}{\partial x_n}e_{x_n}. \end{aligned} \tag{2.10}$$

Erorile relative satisfac relaţia:

$$y \cdot \varepsilon_y = \frac{\partial f}{\partial x_1}x_1 \cdot \varepsilon_{x_1} + \frac{\partial f}{\partial x_2}x_2 \cdot \varepsilon_{x_2} + \cdots + \frac{\partial f}{\partial x_n}x_n \cdot \varepsilon_{x_n}. \tag{2.11}$$

Deoarece în sistemele numerice de calcul, cele mai complicate proceduri se reduc în final la operaţii aritmetice, va fi analizată propagarea erorilor în cazul celor patru operaţii aritmetice elementare:

Adunarea:

$$\begin{aligned} y &= f(x_1, x_2) = x_1 + x_2, \\ e_y &= e_{x_1} + e_{x_2}, \\ \varepsilon_y &= \frac{x_1}{x_1 + x_2}\varepsilon_{x_1} + \frac{x_2}{x_1 + x_2}\varepsilon_{x_2}. \end{aligned} \tag{2.12}$$

Dacă cei doi termeni x_1, x_2 au acelaşi semn, adunarea este o operaţie stabilă numeric, deoarece

$$\left|\frac{x_1}{x_1 + x_2}\right| \leq 1, \quad \left|\frac{x_2}{x_1 + x_2}\right| \leq 1, \quad |\varepsilon_y| \leq |\varepsilon_{x_1}| + |\varepsilon_{x_2}|.$$

Scăderea:

$$\begin{aligned} y &= f(x_1, x_2) = x_1 - x_2, \\ e_y &= e_{x_1} - e_{x_2}, \\ \varepsilon_y &= \frac{x_1}{x_1 - x_2}\varepsilon_{x_1} - \frac{x_2}{x_1 - x_2}\varepsilon_{x_2}. \end{aligned} \tag{2.13}$$

Dacă x_1 şi x_2 au acelaşi semn, operaţia de scădere este instabilă numeric. Eroarea relativă a diferenţei poate depăşi cu multe ordine de mărime eroarea termenilor, prin fenomenul numit de anulare prin scădere, ca în exemplul:

$x_1 = 0,12345 \pm 1\%$; $x_2 = 0,12344 \pm 1\%$; $y = x_1 - x_2 = 0,00001 \pm 3 \cdot 10^4\%$.

Înmulţirea:

$$\begin{aligned} y &= f(x_1, x_2) = x_1 x_2 \\ e_y &= x_2 e_{x_1} + x_1 e_{x_2} \\ \varepsilon_y &= \varepsilon_{x_1} + \varepsilon_{x_2} \end{aligned} \tag{2.14}$$

Înmulţirea este o operaţie stabilă din punct de vedere numeric deoarece

$$|\varepsilon_y| \leq |\varepsilon_{x_1}| + |\varepsilon_{x_2}|.$$

Împărţirea:

$$\begin{aligned} y &= f(x_1, x_2) = \frac{x_1}{x_2}, \\ e_y &= \frac{1}{x_2} e_{x_1} - \frac{x_1}{(x_2)^2} e_{x_2}, \\ \varepsilon_y &= \varepsilon_{x_1} - \varepsilon_{x_2}. \end{aligned} \tag{2.15}$$

Şi împărţirea este o operaţie stabilă din punct de vedere numeric deoarece

$$|\varepsilon_y| \leq |\varepsilon_{x_1}| + |\varepsilon_{x_2}|.$$

În aceste evaluări ale erorilor s-a considerat că toate calculele se efectuează exact. În realitate, rezultatul este rotunjit (la numărul de cifre semnificative specifice sistemului de calcul) şi eroarea relativă a rezultatului este majorată cu eroarea relativă de rotunjire.

În cazul în care prezintă importanţă deosebită evaluarea erorii unui calcul numeric se poate utiliza următorul tip abstract de date care permite controlul propagării erorii. În acest tip de date numerele reale x sunt reprezentate prin valoarea x şi marginea erorii relative Γ_x, deci prin intervalul:

$$x^*(1 - \Gamma_x) \leq x \leq x^*(1 + \Gamma_x). \tag{2.16}$$

real tip inregistrare interval
  real val ;valoare
  real er ;marginea erorii relative
procedura suma (x,y,s) ;calculează s = x + y
  interval x, y, s
  $s.val = x.val + y.val$
  $s.er = err + (|x.val \cdot x.er| + |y.val \cdot y.er|)/|s.val|$
retur
procedura dif(x,y,d) ;calculează d = x - y
  interval x, y, d
  $d.val = x.val - y.val$
  $d.er = err + (|x.val \cdot x.er| + |y.val \cdot y.er|)/|d.val|$
retur
procedura prod(x,y,p) ;calculează p = x * y

```
                    interval x, y, p
                    p.val = x.val · y.val
                    p.er = err + |x.er| + |y.er|
retur
procedura rap(x,y,c)                                  ;calculează c = x / y
                    interval x, y, c
                    c.val = x.val/y.val
                    c.er = err + |x.er| + |y.er|
retur
```

2.2.3 Erori de trunchiere

În esenţă, metodele numerice constau în reducerea rezolvării unei probleme complicate la un număr finit de etape simple, care în fond sunt operaţii aritmetice elementare. Datorită caracterului finit al oricărui algoritm, încercarea de a rezolva probleme de analiză matematică, ce presupun teoretic o infinitate de paşi (cum sunt de exemplu calculele limitelor) determină apariţia unor erori de metodă numite erori de trunchiere.

Calculul numeric al limitelor de şiruri, serii sau funcţii (inclusiv calculul numeric al derivatelor şi integralelor, care se reduc în fond tot la calculul limitelor) presupune trunchierea unui proces numeric infinit. Pentru a evidenţia eroarea de trunchiere se consideră un şir numeric $x_k = 1, 2, \ldots, n$ convergent către limita $x = \lim_{k\to\infty} x_k$. Conform definiţiei convergenţei, pentru orice ε există un n astfel încât $|x - x_k| \leq \varepsilon$, pentru orice $k > n$. În consecinţă, pentru o eroare ε impusă există un rang finit n, astfel încât x_n reprezintă o aproximare satisfăcătoare pentru limita x. Rezultă că, după evaluarea unui numar n suficient de mare de termeni, ultimul poate fi adoptat ca soluţie numerică $x^* = x_n$, cu o eroare de trunchiere dependentă de acest n. În general, eroarea de trunchiere este cu atât mai mică cu cât numărul de termeni calculaţi este mai mare. Modul în care eroarea de trunchiere depinde de n caracterizează viteza de convergenţă a algoritmului. Se spune că un algoritm are *viteza de convergenţă* de ordinul p, dacă există o constantă $c > 0$, astfel încât eroarea de trunchiere satisface inegalitatea:

$$|\varepsilon_n| = |x - x_n| \leq \frac{c}{n^p}, \tag{2.17}$$

pentru orice $n > 1$. Se constată că ordinul vitezei de convergenţă a unui algoritm este dat de panta dreptei ce mărgineşte superior graficul funcţiei $|\varepsilon| = f(n)$ în scară dublu logaritmică.

O categorie importantă de procese numerice infinite o reprezintă seriile Taylor. Acestea, fiind serii de puteri, pot fi utilizate la evaluarea funcţiilor elementare (sinus, cosinus, exponenţială, logaritm, etc.) şi a celor speciale (Bessel, integrale eliptice) prin reducerea la operaţii aritmetice elementare (adunări si înmulţiri).

Pentru a exemplifica modul în care poate fi sumată numeric o serie cu controlul erorii de trunchiere se consideră dezvoltarea în serie Taylor a funcţiei $y = \sin x$:

$$y = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \tag{2.18}$$

Termenul general al seriei poate fi calculat recursiv:

$$T_k = -\frac{(-1)^k}{(2k-1)!}x^{2k-1} = -T_{k-1}\frac{x^2}{(2k-1)(2k-2)}. \tag{2.19}$$

Prin trunchierea seriei la rangul n se obţine

$$y^* = \sum_{k=1}^{n} \frac{(-1)^{k+1}}{(2k-1)!} \cdot x^{2k-1}. \tag{2.20}$$

Deoarece seria este alternantă, eroarea de trunchiere este majorată de modulul primului termen neglijat:

$$|\varepsilon_y| = |y - y^*| \leq \frac{|x|^{2n+1}}{(2n+1)!}. \tag{2.21}$$

Rezultă următorul pseudocod pentru evaluarea funcţiei sinus cu o eroare de trunchiere mai mică decât eroarea de rotunjire.

funcţia sinus(x)
    real x, t, s
    întreg k
    $t = x$
    $s = t$
    $k = 1$
    repetă
        $k = k + 2$
        $t = -tx^2/k/(k-1)$
        $s = s + t$
    până când $|t| <$err
întoarce s

Efortul de calcul pentru evaluarea funcţiei creşte odată cu scăderea erorii impuse, dar şi cu creşterea modulului variabilei x. Deoarece funcţia $\sin(x)$ este periodică, algoritmul poate fi îmbunătăţit prin reducerea variabilei la primul cerc (sau chiar la primul cadran).

2.3 Chestiuni de studiat

1. Determinarea erorii relative de rotunjire a sistemului de calcul;
2. Analiza propagării erorilor inerente;
3. Analiza erorii de trunchiere;
4. Implementarea unor algoritmi numerici care controlează eroarea.

2.4 Modul de lucru

Pentru desfăşurarea lucrării lansaţi lucrarea *Erori în rezolvarea problemelor numerice* din meniul general. Aceasta are ca urmare lansarea următorului meniu:

- Eroarea de rotunjire
- Calcule cu controlul erorii
- Erori de trunchiere

2.4.1 Determinarea erorii relative de rotunjire a sistemului de calcul

Se selectează opţiunea *Eroarea de rotunjire* care are ca efect calculul şi afişarea erorii relative de rotunjire specifice sistemului.

2.4.2 Analiza propagării erorilor inerente

Se selectează opţiunea *Calcule cu controlul erorii* din meniul principal, care are ca efect lansarea unui program ce emulează un calculator de buzunar care foloseşte notaţia poloneză inversă (operand, operand, operator). Acest calculator nu operează cu numere reale ci cu intervale (valoare numerică şi margine superioară a erorii relative).

Programul permite introducerea fie a operanzilor (valoare şi eroare relativă) fie a operatorilor unari (I - invers, O - opus, V - radical) sau binari (+ - * /).

Operanzii sunt introduşi într-o stivă vizualizată pe ecran de 5 coloane (număr de ordine, valoare numerică, eroare relativă, limita minimă şi limita maximă a valorii exacte).

Operatorii unari acţionează asupra ultimului element introdus în stivă, înlocuindu-l cu valoarea obţinută în urma aplicării operatorului (stiva nu îşi modifică dimensiunea).

Operatori binari folosesc ca operanzi ultimele două elemente din stivă, iar rezultatul înlocuieşte aceste elemente (stiva se micşorează cu o unitate).

Pentru a facilita manipularea stivei sunt disponibili încă doi operatori unari, cu caracter nenumeric (R - repetă vârful stivei şi E - extrage element din stivă), care modifică (incrementează respectiv decrementează) înălţimea stivei.

Se propune rezolvarea ecuaţiei de gradul doi:

$$ax^2 + bx + c = 0,$$

în care $a = 1$, $b = -100.01$, $c = 1$ (cu rădăcinile $x_1 = 100$; $x_2 = 0,01$). Valorile a, b, c se vor introduce cu erorile relative de $0.1\% = 0.001$. Se vor nota valorile intermediare:

$$\begin{aligned} \Delta &= b^2 - 4ac, \\ x_1 &= \frac{-b + \sqrt{\Delta}}{2a}, \\ x_2 &= \frac{-b - \sqrt{\Delta}}{2a}, \end{aligned}$$

şi erorile relative asociate.

Pentru a evita erorile datorate anulării prin scădere, se va calcula a doua rădăcină pe baza relaţiei $x_1 x_2 = c/a$

$$x_2 = \frac{c}{ax_1}.$$

Se vor compara erorile relative şi absolute ale rezultatelor obţinute pe cele două căi.

2.4.3 Analiza erorii de trunchiere

Se selectează opţiunea *Erori de trunchiere* din meniul principal, care are ca efect lansarea unui program care realizează sumarea următoarelor serii:

- Taylor pentru funcţia exponenţială
- Taylor pentru funcţia sinus
- Taylor pentru funcţia logaritm natural
- Fourier pentru funcţia crenel

Document disponibil la `www.lmn.pub.ro/Education/methods/methods.html`

- armonica alternantă
- armonica alternantă cu convergenţă îmbunătăţită prin metoda Euler

După selectarea seriei şi alegerea valorii variabilei independente x, programul calculează şi afişează suma parţială a seriei şi eroarea absolută de trunchiere pentru diferite ordine, apoi reprezintă grafic variaţia erorii în funcţie de numărul de termeni sumaţi. Pentru a putea facilita comparaţiile se pot reprezenta şi mai multe grafice de erori simultan.

Se recomandă observarea convergenţei

- seriilor Taylor pentru evaluarea expresiilor:
  - exp(0), exp(1), exp(5), exp(-10), exp(10),
  - sin(0), sin(1), sin(10),
  - ln(1), ln(0.1), ln(1.1).

  Reamintim că dezvoltă'rile în serie Taylor sunt:

$$\begin{aligned} e^x &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \\ \sin(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \\ \ln(x) &= (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \cdots \end{aligned}$$

- seriei Fourie pentru evaluarea expresiilor:
  - crenel(1), crenel(0.1).

  Funcţia crenel este definită astfel

$$\text{crenel}\,(x) = \begin{cases} \frac{\pi}{4} & \text{dacă} \sin(x) > 0 \\ -\frac{\pi}{4} & \text{dacă} \sin(x) \leq 0 \end{cases}$$

  iar dezvoltarea ei în serie Fourier conduce la

$$\text{crenel}\,(x) = \sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5} + \cdots$$

- armonicii alternante

$$A = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \ldots = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{1}{k}$$

- armonicii alternante cu convergenţă îmbunătăţită

$$A = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \ldots = 0.5 + \sum_{k=1}^{\infty} \frac{\left(\frac{1}{2k-1} - \frac{1}{2k+1}\right)}{4k}.$$

Se vor comenta rezultatele şi se va aprecia viteza de convergenţă.

Indicaţie: Pentru estimarea vitezei de convergenţă se calculează panta dreptei ce mărgineşte graficul erorii, în scara dublu logaritmică.

2.4.4 Implementarea unor algoritmi cu controlul erorii

Se va genera pseudocodul funcţiei cosinus, determinată prin serie Taylor trunchiată până la atingerea unei erori de trunchiere impuse. Se va implementa acest algoritm într-un limbaj de programare sub forma unei funcţii cosinus(x, ert), în care x este variabilă independentă iar ert este eroarea de trunchiere impusă. Se va implementa algoritmul de determinare al erorii relative de rotunjire err, iar în final se va implementa algoritmul de evaluare a funcţiei cosinus cu $ert = err$ şi cu controlul propagării erorii, sub forma unei proceduri care să calculeze cos(x) şi eroarea relativă asociată.

2.5 Întrebări şi probleme

1. Cum poate fi caracterizată eroarea unei variabile vectoriale x?
2. Implementaţi algoritmul de determinare a erorii relative de rotunjire în diferite limbaje de programare (în simplă şi dublă precizie) şi apoi comparaţi rezultatele.
3. Ce modificări trebuie aduse algoritmului de determinare a erorii relative de rotunjire pentru ca acesta să determine nu numai ordinul de mărime al erorii ci şi valoarea sa exactă?
4. Definiţi şi implementaţi un tip abstract de date, care să reprezinte fiecare număr real ca o pereche de numere reale ce îl încadrează (valoare maximă, valoare minimă).
5. Este adunarea numerelor reale rotunjite o operaţie asociativă? Pentru a aduna mai multe numere reale diferite cu eroare minimă ele trebuie sortate în ordine crescătoare sau descrescătoare?
6. Pentru a micşora erorile de rotunjire la sumarea unei serii cum trebuie adunaţi termenii, în ordine crescătoare sau descrescătoare?

7. Folosiţi opţiunea *Calcule cu controlul erorii* pentru a determina rezistenţa de şunt ce trebuie folosită la transformarea unui galvanometru de 1mA (în clasa 2%) a cărui rezistenţă $R = 57.5\Omega$ a fost determinată cu precizie de 0,5(%) într-un ampermetru de 1A.

8. Generaţi un algoritm pentru rezolvarea unei ecuaţii de gradul doi, care evită fenomenul de anulare prin scădere.

9. Generaţi şi implementaţi un algoritm, pentru evaluarea funcţiilor Bessel.