Unitatea de învăţare 2 - *2 ore*

Precizia calculelor numerice. Aplicaţii – *continuare*

2.1. Reprezentarea numerelor în virgulă mobilă
2.2. Operaţii în virgulă mobilă
2.3. Erori de rotunjire

Cunoştinţe şi deprinderi

La finalul parcurgerii acestei unităţi de învăţare vei înţelege:

- exprimarea numerelor în diferite baze de numeraţie şi conversia dintr-o bază în alta;
- efectuarea operaţiilor în virgulă mobilă;
- ce sunt erorile de rotunjire.

În această unitate de învăţare sunt prezentate concepte privind erorile, exemple ale acestora, precum si reprezentarea numerelor in diferite baze de numeratie si conversia acestora intre baze.

2.1. Reprezentarea numerelor în virgulă mobilă

<u>Exprimarea numerelor în baza zece</u>. În general suntem obişnuiţi să gândim şi să calculam cu numere în baza 10. Astfel, numărul zecimal 257,424 poate fi scris sub forma

$$2 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0 + 4 \cdot 10^{-1} + 2 \cdot 10^{-2} + 4 \cdot 10^{-3}.$$

Orice număr real în baza 10 are o reprezentare unică în forma preyentată anterior. Se exceptează cazul în care, în scrierea poziţională a unui număr zecimal, cifra 9 apare de o infinitate de ori. De exemplu numărul 99,9999 ... reprezintă acelaşi număr ca 100.

În general, pentru reprezentarea unui număr real putem utiliza o bază oarecare *B*, $B \geq 2$, *B* fiind un număr natural. În această bază de numeraţie, orice numar real pozitiv admite o reprezentare unică de forma

$$a_n \cdot B^n + a_{n-1} \cdot B^{n-1} + ... + a_1 \cdot B + a_0 + a_{-1} \cdot B^{-1} + a_{-2} \cdot B^{-2} + ...$$

unde coeficienţii *a* se numesc cifre în *sistemul cu baza B* şi sunt numere întregi pozitive cu proprietatea

$$0 \leq a_i \leq B - 1.$$

Avantajul scrierii unui număr într-o bază care conduce la o scriere poziţională de forma [4]

$$a_n a_{n-1} ... a_1 a_0, a_{-1} a_{-2} ...,$$

unde $a_n a_{n-1} ... a_1 a_0$ reprezintă *partea întreagă*, iar 0, $a_{-1} a_{-2} ...$ *partea fracţionară* a numărului real, este acela că pentru operaţiile aritmetice putem utiliza reguli simple. Cu cât baza este un număr natural mai mic, cu atât aceste reguli sunt mai simple. Acest argument stă la baza faptului că cele mai multe calcule opereaza în baza 2, respectiv în sistemul binar cu două cifre: 0 şi 1.

<u>Conversia în binar a unui număr întreg</u>. Pentru conversia unui număr întreg N_I, exprimat în baza **10**, într-o altă bază **b**, se foloseşte metoda împărţirii repetate, coform urmatorului algoritm:

a) se împarte numărul întreg N_I prin baza **b**, obţinându-se cătul Q_1 şi restul R_0;

b) se împarte câtul Q_1 prin baza **b**, obţinându-se câtul Q_2 şi restul R_1;

c) se continuă operaţia de la punctul b până când se obţine $Q_n = 0$.

Resturile obţinute, asezate în ordine inversă a apariţiei lor, reprezintă cifrele numărului convertit în baza **b**, R_0 fiind cifra cea mai puţin semnificativă [3].

<u>Exemplul 1</u>. Să se reprezinte în baza $b = 2$ numărul zecimal 75.

<u>Rezolvare</u>. În tabelul 2.1 sunt prezentate rezultatele algoritmului "împărţirii repetate". În consecinţă, numărul în binar, obţinut din numărul zecimal 75, este **1001011.** Conversia din binar în zecimal a numărului 1001011 este realizată astfel:

$$N = 1 * 2^6 + 0 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 =$$

$= 64 + 8 + 2 + 1 = 75$.

Având în vedere rezultatul obţinut, rezultă că procesul de conversie a unui număr întreg în binar decurge fără erori de conversie.

Tabelul 2.1

Rezultatele obţinute la conversia în binar a numărului 75_{10}

Numărul sau câtul de împărţit	Câtul obţinut	Restul obţinut
75	37	1
37	18	1
18	9	0
9	4	1
4	2	0
2	1	0
1	0	1

Conversia în binar a unui număr fracţionar. Pentru conversia unui număr zecimal subunitar N_s, într-o altă bază **b**, se procedează în modul următor:

- se înmulţeşte numărul zecimal subunitar dat N_s cu baza **b**, rezultând partea fracţionara F_1 şi cifra A_1;
- se înmulţeşte partea fracţionară F_1 cu baza **b**, obţinându-se partea fracţionara F_2 şi cifra A_2;
- se continuă înmulţirea până când partea fracţionară F_m este egală cu 0 sau până când se obţine un număr de cifre A_i apropiat de precizia cu care se doreşte să se reprezinte partea fracţionara a numărului N_s.

Şirul $A_1A_2A_3 \dots A_m$ formează numărul convertit în baza **b**, A_1 fiind cifra cea mai semnificativă. Dacă după un număr finit de înmulţiri succesive se ajunge la o parte fracţionară $F_k = 0$, atunci reprezentarea numărului zecimal N_s în baza **b** va avea un număr finit de cifre. În caz contrar numărul de cifre va fi infinit [1].

Exemplul 2. Să se reprezinte numărul zecimal 0,73 în baza $b = 2$.

Rezolvare. În tabelul 1.3 sunt prezetate rezltatele algoritmul "înmulţirii repetate".

Tabelul 2.2

Rezultatele obţinute la conversia în binar a numărului $0{,}73_{10}$

Numărul sau partea fracţionară	Numărul obţinut	Partea fracţionară obţinută	Cifra binară obţinută
0,73	1,46	0,46	1
0,46	0,92	0,92	0
0,92	1,84	0,84	1
0,84	1,68	0,68	1
0,68	1,36	0,36	1

În consecinţă, numărul obţinut în binar, exprimat cu cinci cifre zecimale, este **0,10111.**

Conversia din binar în zecimal a numărului 0,10111 este dată de relaţia

$$N = 1*2^{-1} + 0*2^{-2} + 1*2^{-3} + 1*2^{-4} + 1*2^{-5} =$$

$$= 0{,}5 + 0{,}125 + 0{,}0625 + 0{,}03125 = 0{,}71875 .$$

<u>Concluzie</u>. Conversia în binar a numerelor zecimale este însoţită de erori. În cazul concret al conversiei în binar a numărului 0,73 cu cinci zecimale, eroarea este

$$e = 0{,}73 - 0{,}71875 = 0{,}01125 ,$$

respectiv

$$\varepsilon = \frac{e}{N} = \frac{0{,}01125}{0{,}73} * 100 = 1{,}54\% .$$

<u>Conversia în binar a numerelor reale</u>. În cazul numerelor reale, operaţia de conversie a acestor numere într-o bază de numeraţie ***b*** se reduce la reprezentarea unui număr întreg $\boldsymbol{N_I}$ şi a unui număr subunitar $\boldsymbol{N_S}$, conform relaţiei

$$\mathbf{R = N_I + N_S} .$$

Pentru numere fracţionare se utilizează reprezentarea internă virgulă mobilă. Conform standardului **IEEE** (Institute for Electrical and Electronics Engineers) datele se memoreaza pe 32 de biti în simplă precizie şi pe 64 de biţi în dublă precizie. Pentru reprezentari în simplă precizie, macheta datelor este prezentată în figura 1.1.

Fig. 2.1.

s	Caracteristica	Fractie
31	30 . . . 23	22 21 20 . . . . 2 1 0

Reprezentarea internă în virgula mobilă.

Aceasta macheta presupune ca numărul de reprezentat are următoarea exprimare binară

$$\mathrm{m} = (-1)^s * 1, fractie * 2^{\exp onent} \tag{2.1}$$

unde **s** este valoarea bitului de semn (1 pentru mantisa negativă şi 0 pentru mantisa pozitivă) iar **fractie** reprezintă partea fracţionară a mantisei.

Mantisa este normalizată şi are întotdeauna forma **1,fractie**, ceea ce înseamnă că are valori în intervalul [1, 2). Pentru ca mantisa să fie adusă la această formă se modifică în mod corespunzător exponentul numarului **m** [7].

<u>*Exemplul 3.*</u> Fie un număr binar **m = 101,0111**. Să se normalizeze numărul *m*.

<u>*Rezolvare*</u>. Scrierea numărului ***m*** în formă normalizată este $m = 1{,}010111 * 2^2$.

<u>*Observaţie*</u>. Deoarece partea întreaga a mantisei este întotdeauna 1, aceasta nu se reprezintă intern. **Fractia** se reprezintă intern sub forma semn-mărime (semnul exprimat prin bitul de semn iar mărimea exprimată în cod direct). Pentru a nu se utiliza doi biţi de semn, unul pentru mantisă şi unul pentru exponent, convenţia IEEE a înlocuit exponentul din macheta de reprezentare cu o **caracteristica**. Aceasta este o valoare în exces faţă de 127, pentru reprezentări în simplă precizie, respectiv 1023 pentru reprezentări în dublă precizie

caracteristica = exponent + 127 (simplă precizie);

caracteristica = exponent + 1023 (dublă precizie).

<u>*Exemplul 4.*</u> Să se reprezinte în virgulă mobilă, în simplă precizie, numărul zecimal **75,73.**

<u>*Rezolvare.*</u> Numarul zecimal N = 75,73 se descompune in partea intreaga,

N_I = 75 si partea fractionara N_Z = 0,73

$$\mathbf{N = N_I + N_Z}$$

75,73 = 75 + 0,73.

Reprezentarea in cod binar direct a numarului N_I este prezentata in exemplul 3.1, rezultatul fiind **N_I = 1001011**. Reprezentarea partii fractionare N_z este tratata in exemplul 3.2, rezultatul partial al conversiei fiind **N_z = 0,10111...** .Prin insumarea celor doua componente ale numarului N se obtine

N = 1001011 + 0,10111 = 1001011,10111

Mantisa normalizata a numarului binar N va fi

N = 1001011,10111 = 1,00101110111 x 2^6 .

Fractia asociata mantisei normalizate are valoarea

fractie = 00101110111

iar caracteristica va fi

caracteristica = 6 + 127 = 133

respectiv in cod direct

caracteristica = 10000101.

Reprezentarea interna pentru numarul 75,73 este

31	30 . . . 23	22 . . . . 2 1 0
0	1 0 0 0 0 1 0 1	0 0 1 0 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

Caracteristica are urmatoarele valori normale:

0 < caracteristica < 255 (simpla precizie);

0 < caracteristica < 2047 (dubla precizie).

Cand caracteristica are valoarea **0**, numarul reprezentat intern este zero. Cand caracteristica este **255** (respectiv 2047) se considera **depasire in virgula mobila**.

Caracteristicile reprezentarilor interne in virgula mobila sunt prezentate in tabelul 2.3.

Tabelul 2.3

Caracteristici ale datelor reprezentate in virgula mobila

Caracteristici	**Tip reprezentare**	
	Simpla precizie	Dubla precizie
Numar biti pentru reprezentare caracteristica	**8**	**11**
Numar biti pentru reprezentare fractie	**23**	**52**
Valoare minima caracteristica	**1**	**1**
Valoare maxima caracteristica	**254**	**2047**
Eroare maxima fractie	$2^{-24} \approx 10^{-7}$	$2^{-53} \approx 10^{-16}$
Cel mai mic numar pozitiv	$2^{1-127} \approx 10^{-38}$	$2^{1-1023} \approx 10^{-307}$
Cel mai mare numar pozitiv	$2\text{x } 2^{254-127} \approx 10^{38}$	$2\text{x } 2^{2047-1023} \approx 10^{307}$
Domeniu de reprezentare	-10^{38} ... 10^{38}	-10^{307} ... 10^{307}

În sistemele de calcul este folosită în mod curent reprezentarea în virgulă mobilă pentru numerele reale.

Virgula mobilă reprezintă forma numerelor reale exprimate printr-o fracţie, numită *mantisă*, şi un număr întreg cu funcţie de exponent, numit *caracteristică*. Dacă se consideră *b* baza sistemului de numeraţie, *f* – fracţia şi *e* – exponentul, atunci un număr exprimat în virgulă mobilă are forma $f * b^e$. De exemplu, în sistemul zecimal, numărul 574,12 va fi reprezentat sub forma $0{,}57412*10^3$ [1].

Normalizarea numerelor reprezintă operaţia prin care forma numărului va avea prima cifră a mantisei diferită de zero. Un număr reprezentat în sistemul zecimal prin virgulă mobilă are forma $x = f * 10^e$. Dacă se utilizează forma normalizată, *f* nu poate fi mai mic de 0,1, prima cifră fiind astfel diferită de zero.. Deoarece *f* este o fracţie, valoarea sa absolută nu poate fi mai mare ca 1. În consecinţă

$$0{,}1 \le f < 1 . \qquad (2.2)$$

Calculatoarele operează cu numere reale, rezervând un anumit număr de cifre pentru mantisa *f* şi un altul pentru exponentul *e*. De exemplu, pentru unele medii de programare, reprezentarea numerelor reale în virgulă mobilă este realizată utilizând 32 de biţi, din care 24 sunt pentru mantisa, 7 pentru caracteristică şi un bit pentru semnul numărului, figura 1.1.

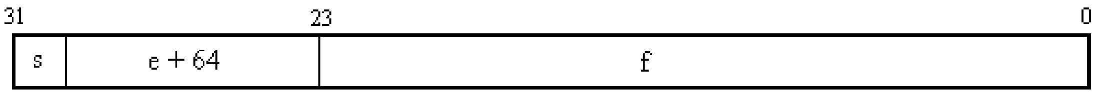

Fig. 2.2. Reprezentarea datelor de tip real într-o locaţie de 32 bit.

Exemplul 5. Să se normalizeze următoarele numere: 542,36; 4527; 0,0002617; –42,15; –0,012. Soluţia este prezentată în tabelul 2.4.

Tabelul 2.4.

Numărul	Numărul normalizat	Mantisa	Exponentul
542,36	$0{,}54236 \cdot 10^3$	0,54236	3
4527	$0{,}4527 \cdot 10^4$	0,4527	4
0,0002617	$0{,}2617 \cdot 10^{-3}$	0,2617	–3
–42,15	$-0{,}4215 \cdot 10^2$	-0,4215	2
–0,012	$-0{,}12 \cdot 10^{-1}$	–0,12	–1

2.2. Operaţii în virgulă mobilă

Fie două numere, $x_1 = 124{,}7012$ şi $x_2 = 0{,}01028$. Adunarea acestor două numere conduce la rezultatul $x_3 = 124{,}71148$.

Se consideră un calculator cu 7 cifre pentru mantisă. Pentru a aduna cele două numere cu acest calculator, numerele trebuiesc întâi normalizate, obţinându-se formele $x_1 = 0{,}1247012 \cdot 10^3$ şi $x_2 = 0{,}1028 \cdot 10^{-1}$. Operaţia de adunare necesită alinierea virgulei, prin deplasare spre dreapta a mantisei lui x_2, astfel încât exponentul acestuia să fie egal cu exponentul primului număr

$$x_3 = x_1 + x_2 = 0{,}1247012 \cdot 10^3 + 0{,}00001028 \cdot 10^3 = 0{,}12471148 \cdot 10^3.$$

Acest rezultat poate fi pus sub forma a doi termeni normalizaţi, primul având 7 cifre pentru mantisă iar cel de al doilea un exponent cu 7 unităţi mai mic decât primul

$$x_3 = 0{,}1247114 \cdot 10^3 + 0{,}8 \cdot 10^{-4}. \tag{2.3}$$

Pentru cazul general se presupune ca mantisa numerelor reprezentate în virgulă mobilă conţine *t* cifre. Oricare din cele patru operaţii aritmetice va conduce la un rezultat care poate fi descompus în doi termeni

$$x = f \cdot 10^{e} + g \cdot 10^{e-t}\,, \tag{2.4}$$

unde f îndeplineşte restricţiile (2.2).

2.3. Erori de rotunjire

Din exemplul precedent, s-a constatat că în cadrul operaţiilor aritmetice în virgulă mobilă, rezultatul obţinut nu poate fi memorat în mod corespunzător şi în consecinţă vor apare erori.

Eroarea de rotunjire reprezintă eroarea realizată prin modul în care se ia în consideraţie mantisa *g* în componenţa mantisei *f* a rezultatului (1.11). Dintre procedeele de rotunjire sunt de subliniat:

- rotunjirea prin tăiere ;
- rotunjirea simetrică ;
- rotunjirea uniformă.

Rotunjirea prin tăiere presupune stabilirea aproximatei $\bar{x}$ prin neglijarea lui g

$$\bar{x} = f \cdot 10^{e}\,. \tag{2.5}$$

De exemplu, dacă se aplică rotunjirea prin tăiere pentru x_3 din calculul (2.3), se obţine

$$\bar{x}_3 = f \cdot 10^{e} = 0{,}1247114 \cdot 10^{3}\,.$$

Eroarea relativă a rotunjirii prin tăiere este dată de

$$\varepsilon_x = \left|\frac{a_x}{\bar{x}}\right| = \left|\frac{g \cdot 10^{e-t}}{f \cdot 10^{e}}\right| \le \frac{1 \cdot 10^{e-t}}{0{,}1 \cdot 10^{e}} = 10^{1-t}\,. \tag{2.6}$$

Eroarea relativă creşte cu valoarea mantisei *g* şi scade cu valoarea mantisei *f*. Eroarea relativă maximă a rezultatului operaţiilor aritmetice în virgulă mobilă nu depinde de mărimea numerelor, ci numai de *t*, adică de numărul cifrelor mantisei.

Rotunjirea simetrică este definită prin relaţia

$$\bar{x} = \begin{cases} f \cdot 10^{e}\,,\ daca\ |g| < 1/2 \\ f \cdot 10^{e} \pm 10^{e-t}\,,\ daca\ |g| \ge 1/2 \\ 0{,}0\ldots 0 \cdot 10^{-9\ldots 9}\,,\ daca\ x = 0 \end{cases}. \tag{2.7}$$

În cazul rezultatului (2.3) se observă că $g = 0{,}8 > 1/2$, astfel încât aplicarea rotunjirii simetrice conduce la

$$\bar{x}_3 = 0{,}1247114 \cdot 10^3 + 10^{3-7} = 0{,}1247114 \cdot 10^3 + 0{,}0000001 \cdot 10^3 = 0{,}1247115 \cdot 10^3 .$$

Eroarea relativă este

$$\varepsilon_x = \left|\frac{a_x}{\bar{x}}\right| = \left|\frac{(1/2)\cdot 10^{e-t}}{f \cdot 10^e}\right| \le \left|\frac{(1/2)\cdot 10^{e-t}}{0{,}1 \cdot 10^e}\right| = \frac{1}{2}\cdot 10^{1-t} . \tag{2.8}$$

Test de autoevaluare

1. *Scrieti în bază 10 numerele: $(1000)_2$, $(100000)_8$, $(A64)_{16}$.*
2. *Un calculator cu două poziţii zecimale pentru exponent şi 4 poziţii zecimale pentru mantisă. Să se efectueze în reprezentarea virgulă mobilă, cu rotunjire, următoarele operaţii:*

   $2{,}1350 \cdot \pi$; $0{,}2567 \cdot e$; $\pi \cdot e$; 0,2125-0,212 ; 1,2137+22,2137.
3. *In cazul reprezentării în calculator a numărului 1/3, (indiferent pe cîte poziţii zecimale (în număr finit)), coincide eroarea de rotunjire cu eroarea de trunchiere ?*

Lucrare de verificare

1. **Ce sunt erorile de trunchiere. Dati exemple de astfel de erori.**
2. **Definiti erorile absolute si relative.**
3. **Cum se realizeaza conversia in binar a unui numar intreg?**
4. **Definiti normalizarea numerelor. Dati un exemplu de numar normalizat, precizand mantisa si exponentul.**