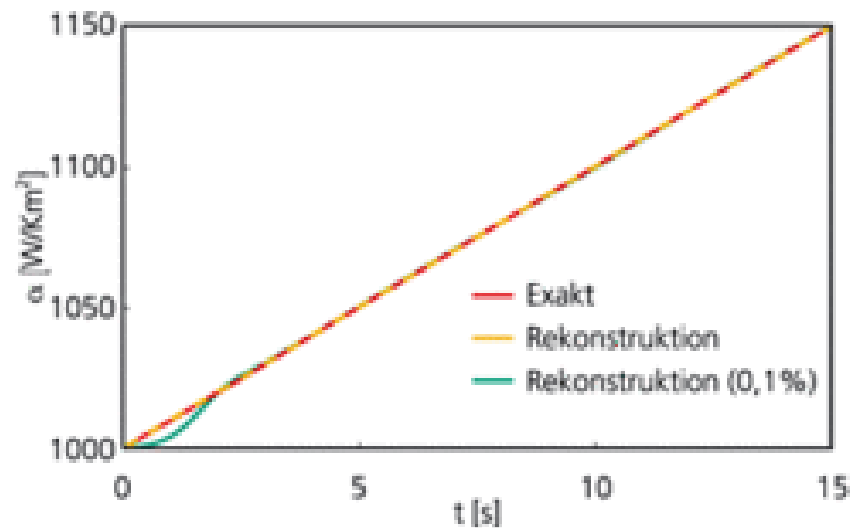
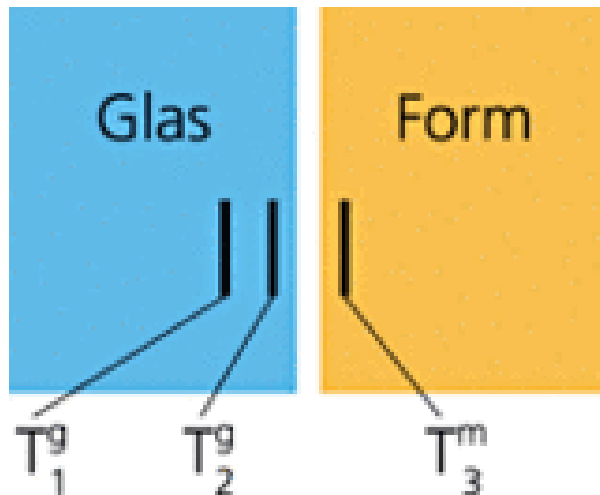


# Prelucrarea datelor experimentale

## Exemplu (ITWM Kaiserslautern)

- Identificarea coeficientului transferului de căldură dintre sticla topită și forma solidă în care este turnată:

$$-kA \frac{\partial}{\partial y} (T - T_s)|_{y=0} = hA (T_s - T_\infty)$$



## Exemplu (AQTR 2008)

Dererminarea puterii anuale a unei turbine eoliene:

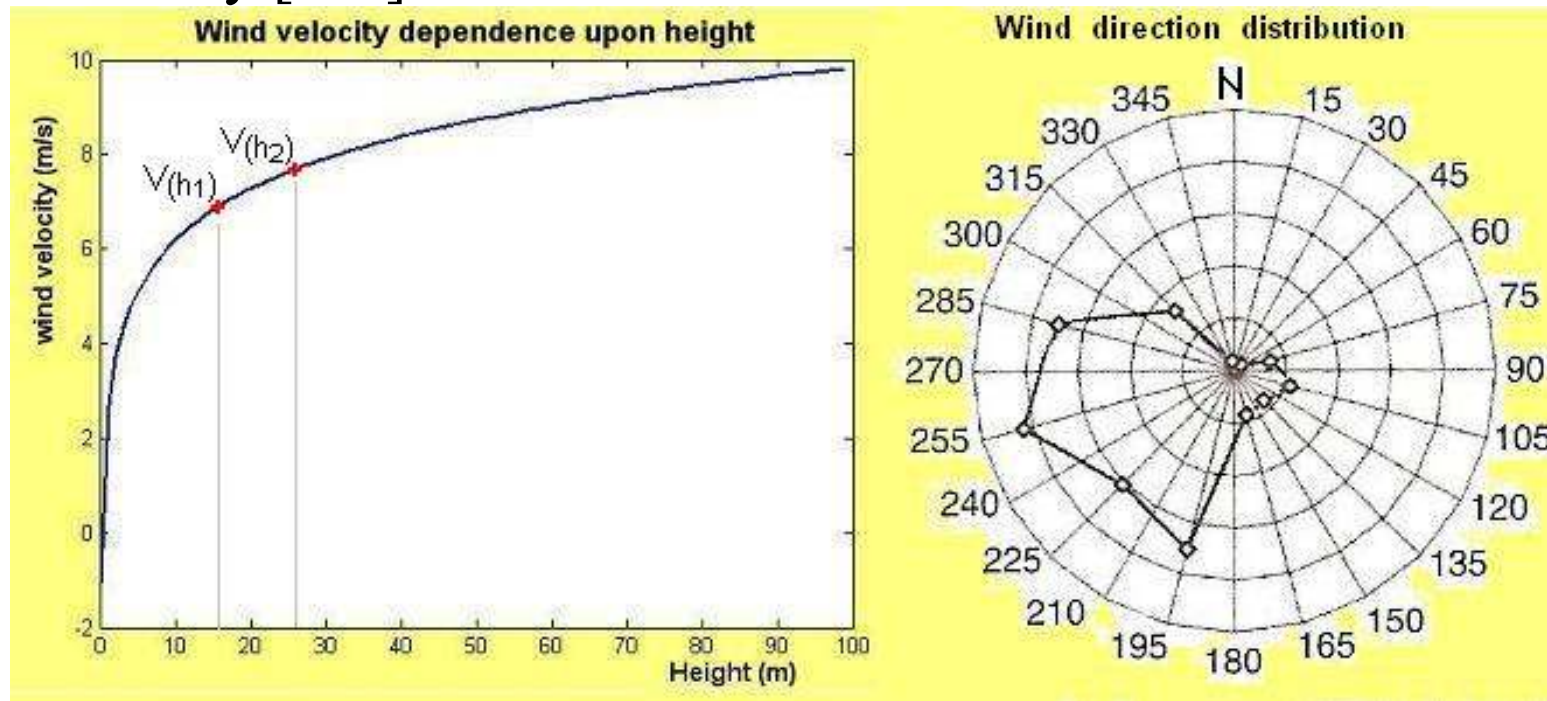
$$P_{vant} [W] = 0.5 \cdot \rho_{aer} \cdot S \cdot v^3$$

$P_{vant}$  – total wind power pass through swept area  $S$

$\rho_{aer}$  – airdensity [ $\text{kg}/\text{m}^3$ ]

$S$  – swept area of rotor wings [ $\text{m}^2$ ]

$v$  – wind velocity [ $\text{m}/\text{s}$ ].



## **Obiective**

Inițirea în problematica și metodele de prelucrare a datelor experimentale

## **Conținutul disciplinei**

- Erori. Măsurare și clasificare.
- Interpolarea datelor (polinomială, spline). Calculul eficient al polinoamelor de interpolare.
- Metoda celor mai mici pătrate.
- Regresie liniară. Modelul liniar. Predicție. Inferențe asupra coeficienților și modelului. Potrivirea curbelor
- Modele liniare generalizate
- Elemente de statistică multidimensională
- Metode numerice
- Vizualizarea datelor - grafică 2D și 3D. Tehnici de vizualizare a volumelor.

## **Bibliografie**

- P. Blaga - Statistică prin ... MATLAB, Presa Universitară Clujeana, Cluj-Napoca, 2003
- D.Ciurchea, V.Chiș - Prelucrarea datelor experimentale, Litografia UBB, Cluj-Napoca, 1995.
- R. Trîmbițaș Analiză numerică.O introducere bazată pe MATLAB, Presa Universitară Clujeană, 2005
- R. Trîmbițaș - Metode statistice, Presa Universitara Clujeana, Cluj-Napoca, 2000

## **Suport**

[www.math.ubbcluj.ro/~tgrosan](http://www.math.ubbcluj.ro/~tgrosan)

## **Cerințe**

- Predarea tuturor laboratoarelor – în fiecare saptamana se va primi un laborator ce are termenul de predare doua saptamani. Întârziarea cu o saptamana scade nota acordată laboratorului cu 1 punct.
- Proiect final (fiecare student va primi o temă pe care o va rezolva, redacta și prezenta)
- Nota finala: 50% laboratoare +50% proiect

# Erori

## 1. Introducere

În prelucrarea datelor provenite din experimente sa apară diferite tipuri de erori:

### 1) Erori în datele de intrare

- a) **Erori personale sau greșeli** – neatenția citirii unui instrument, poziționarea ochiului, eroare de calcul, etc
- b) **Erori sistematice** – legate de calibrarea instrumentelor sau a tehnicii de utilizare a acestora
- c) **Erori aleatoare** – din cauza fluctuațiilor de căldură, tensiune, etc

**2) Erori de rotunjire** – restrangerea numărului de cifre cu care lucrăm

**3) Erori de aproximare.** Dintre acestea amintim:

a) **Erori de trunchiere:**

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!},$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}.$$

## b) Erori de discretizare

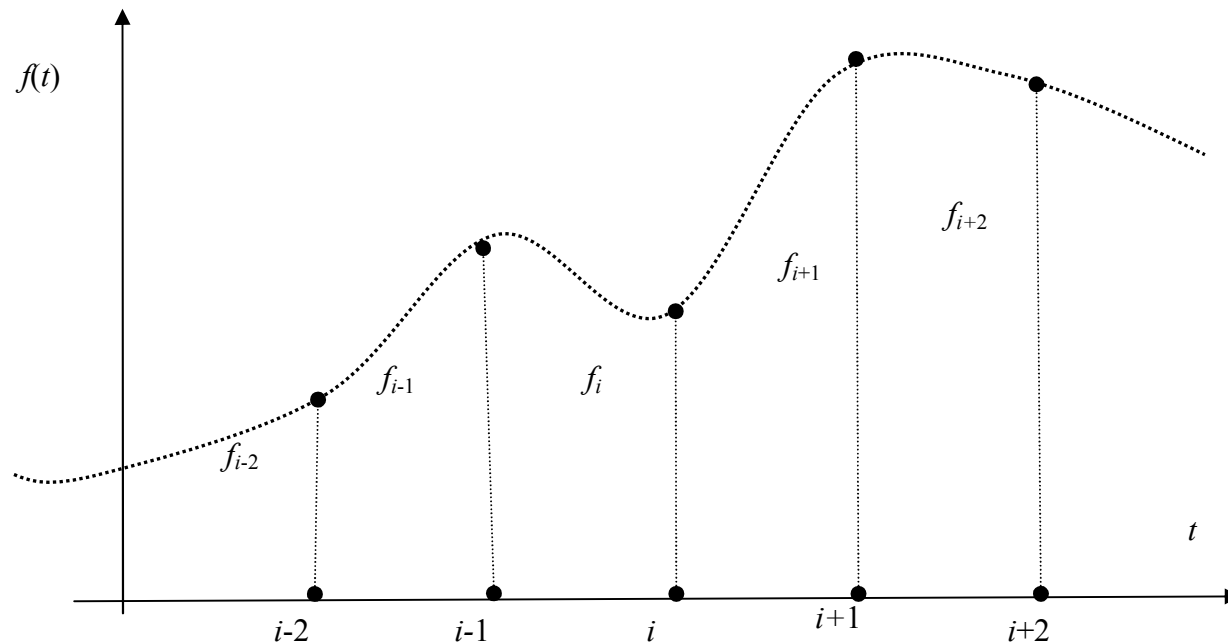
Exemplu:

Fie ecuația diferențială:

$$f'' + f' + f^2 = 0$$

$$f(0) = 0, \quad f(1) = 0$$

Pentru a rezolva numeric introducem o rețea de puncte



Aproximăm derivatele prin

$$f''(x) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{\Delta x^2}; \quad f'(x) \approx \frac{f(x_{i+1}) - f(x_{i-1}))}{2\Delta x}$$

Problema se reduce la rezolvarea sistemului de ecuații

$$f_1 = 0$$

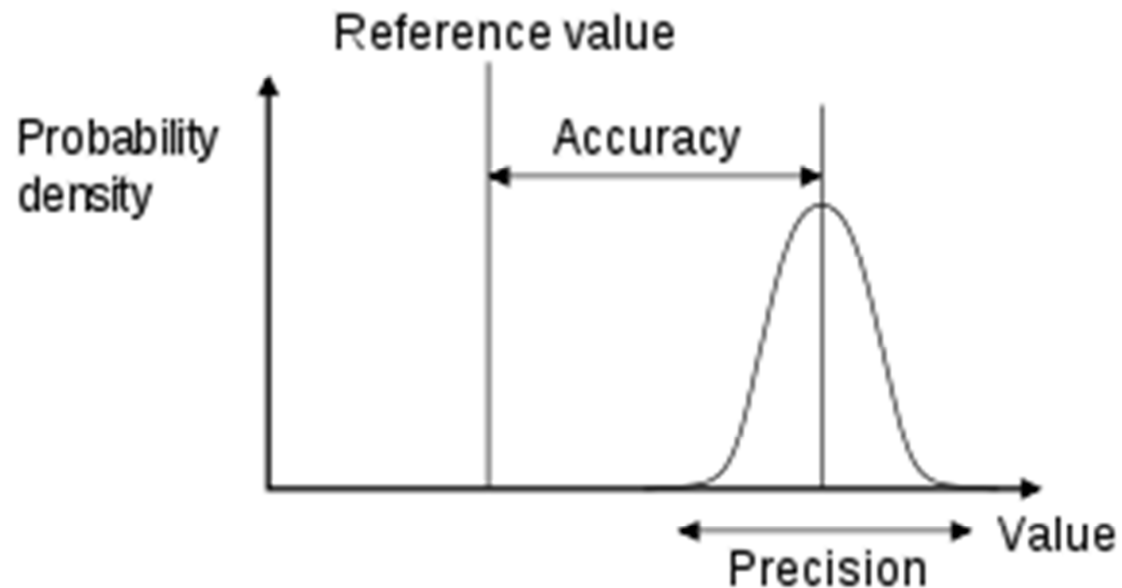
$$\frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + \frac{f_{i+1} - f_{i-1}}{2\Delta x} + f_i^2 = 0, \quad i = 2 : N - 1$$

$$f_N = 0$$

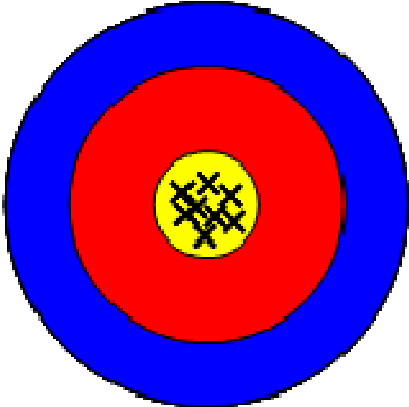
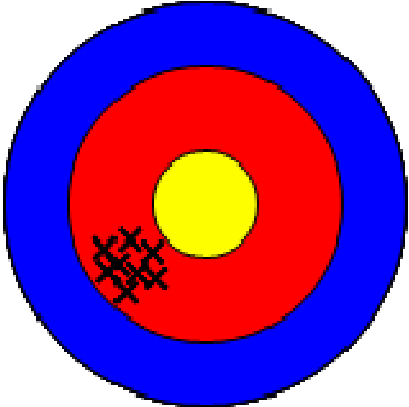
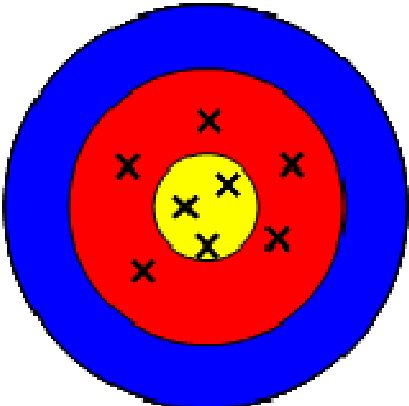
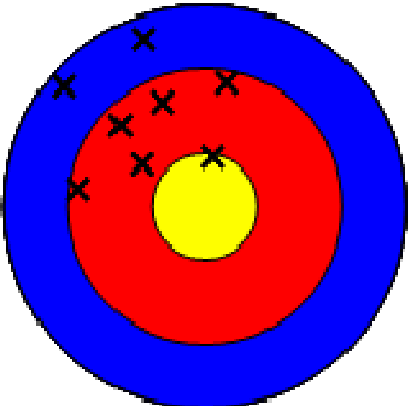
Pe lângă erorile introduse anterior (discretizare) se mai introduc erorile datorate algoritmului de rezolvare a sistemului neliniar de ecuații.



Datele obținute experimental se mai caracterizează și prin **acuratețe** și **precizie**. Dacă datele obținute experimental sunt grupate vom spune că acestea sunt precise, iar dacă sunt apropiate de valoarea pe care o aproximează vom spune că sunt caracterizate de acuratețe.



# Exemplu

	Accurate	Inaccurate (systematic error)
Precise		
Imprecise (reproducibility error)		

## 2. Măsuri ale erorii

**Definiție.** Fie  $X$  un spațiu liniar normat și  $x \in X$  și  $A \subset X$ . Elementul  $x^* \in A$  se numește aproximantă a lui  $x$  din  $A$ . (notăm cu  $x^* \approx x$ ).

**Definiție.** Fie  $x^*$  o aproximantă a lui  $x$ . Atunci

$\Delta x = x - x^*$  - se numește eroare

$\|\Delta x\| = \|x - x^*\|$  - se numește eroare absolută

**Definiție.** Mărimea

$\delta x = \frac{\|\Delta x\|}{\|x\|}$ ,  $x \neq 0$  - se numește eroare relativă

În practică mărimea  $x$  este de obicei necunoscută și atunci se utilizează pentru eroarea relativă expresia:  $\delta x = \frac{\|\Delta x\|}{\|x^*\|}$ .

În practică măsurătorile se fac de obicei cu ajutorul instrumentelor dotate cu o scală gradată cu unități și subunități de măsură specifice. Se poate considera că eroare de citire este un număr egal cu  $\pm \frac{1}{2}$  din subdiviziune.

- ✓ Dacă se măsoară cu metrul se poate presupune o eroare de  $\pm 0.5$  mm.
- ✓ Pentru măsurarea unui unghi cu raportorul se acceptă o eroare de  $\pm 0.5^\circ$ .
- ✓ Dacă se fac măsurători de timp, atunci pornirea și oprirea cronometrului necesită  $\pm 0.25$  secunde.
- ✓ În măsurătorile din electricitate se acceptă abateri de  $\pm 3\%$  din valoarea măsurată.

### 3. Propagarea erorilor

Fie două mărimi A și B măsurate sau obținute cu erorile  $\Delta A$  și  $\Delta B$

#### Adunarea

$$C = A + B$$

$$C \pm \Delta C = (A \pm \Delta A) + (B \pm \Delta B) = (A + B) \pm (\Delta A + \Delta B)$$

$$\Delta C = \Delta A + \Delta B$$

#### Scăderea

$$C = A - B$$

$$C \pm \Delta C = (A \pm \Delta A) - (B \pm \Delta B) = (A - B) \pm (\Delta A + \Delta B)$$

$$\Delta C = \Delta A + \Delta B$$

## Înmulțirea

$$C = (A) * (B)$$

$$C \pm \Delta C = (A \pm \Delta A) * (B \pm \Delta B) = A * B \pm B * \Delta A \pm A * \Delta B \pm \Delta A * \Delta B$$
$$\pm \Delta C = \pm (B * \Delta A + A * \Delta B + \Delta A * \Delta B)$$

sau

$$\pm \Delta C = \pm (A * B) (\Delta A / A + \Delta B / B + \Delta A * \Delta B / A * B)$$

Dacă  $\Delta A / A \ll 1$  și  $\Delta B / B \ll 1$  atunci  $(\Delta A * \Delta B) / (A * B) \rightarrow 0$ .

Atunci

$$C = A * B \text{ și } \Delta C = (A * B) (\Delta A / A + \Delta B / B)$$

## Împărțirea

Se arată analog că

$$C = A / B \text{ și } \Delta C = (A / B) (\Delta A / A + \Delta B / B)$$

# Operații complexe

(R. Trîmbițaș, 2005, *Analiza numerică. O introducere bazată pe MATLAB*, Presa Universitară Clujeană)

Fie  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $x = (x_1, \dots, x_n)$  și  $x^* = (x_1^*, \dots, x_n^*)$ . Dorim să evaluăm eroarea absolută și relativă  $\Delta f$  și respectiv  $\delta f$  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  $f$ . Să presupunem că  $x = x^* + \Delta x$ , unde  $\Delta x = (\Delta x_1, \dots, \Delta x_n)$ . Pentru eroarea absolută avem (folosind formula lui Taylor)

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

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

Dacă  $\Delta x_i$  sunt suficient de mici, atunci  $\Delta x_i \Delta x_j$  sunt neglijabile comparativ cu  $\Delta x_i$  și obținem

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


**EXAMPLE 6** The dimensions of a rectangular box are measured to be 75 cm, 60 cm, and 40 cm, and each measurement is correct to within 0.2 cm. Use differentials to estimate the largest possible error when the volume of the box is calculated from these measurements.

**SOLUTION** If the dimensions of the box are  $x$ ,  $y$ , and  $z$ , its volume is  $V = xyz$  and so

$$dV = \frac{\partial V}{\partial x} dx + \frac{\partial V}{\partial y} dy + \frac{\partial V}{\partial z} dz = yz dx + xz dy + xy dz$$

We are given that  $|\Delta x| \leq 0.2$ ,  $|\Delta y| \leq 0.2$ , and  $|\Delta z| \leq 0.2$ . To find the largest error in the volume, we therefore use  $dx = 0.2$ ,  $dy = 0.2$ , and  $dz = 0.2$  together with  $x = 75$ ,  $y = 60$ , and  $z = 40$ :

$$\begin{aligned}\Delta V \approx dV &= (60)(40)(0.2) + (75)(40)(0.2) + (75)(60)(0.2) \\ &= 1980\end{aligned}$$

Thus, an error of only 0.2 cm in measuring each dimension could lead to an error of as much as 1980 cm<sup>3</sup> in the calculated volume! This may seem like a large error, but it's only about 1% of the volume of the box. 

(Calculus, 5th ed., J. Stewart)



**Exemplu:** Calculați numărul de moli conținuți într-un gaz perfect menținuti într-un recipient care măsoară 1L cu o precizie de 0,5% la o presiune  $P=1 \text{ atm}$ , determinată la o precizie de 1% și termostatul de 300K ce poate fi reglat cu 1/10 grade. Se dă constanta  $R=0,08205 \text{ atm L mol}^{-1}\text{k}^{-1}$ .

*Soluție:* Numărul de moli este dat de  $N = \frac{PV}{RT}$ .

Înlocuind cu datele din problemă avem  $N = \frac{1}{300 \cdot 0,08205} = 0,04062 \text{ mol}$

Înainte de a estima eroarea, calculăm diferențiala

$$dN = \frac{\partial N}{\partial P} dP + \frac{\partial N}{\partial T} dT + \frac{\partial N}{\partial V} dV = \frac{V}{RT} dP - \frac{PV}{RT^2} dT + \frac{P}{RT} dV$$

Folosind suma valorilor absolute

$$|\sum dN| \leq \left| \sum \frac{V}{RT} dP - \frac{PV}{RT^2} dT + \frac{P}{RT} dV \right| \Leftrightarrow \Delta N \leq \left| \frac{V}{RT} \right| \Delta P + \left| \frac{PV}{RT^2} \right| \Delta T + \left| \frac{P}{RT} \right| \Delta V$$

$$\Leftrightarrow \Delta N \leq 0,04062 \cdot 0,01 + 0,0001354 \cdot 0,1 + 0,04062 \cdot 0,005 = 0,000622 \text{ mol}.$$

Pentru a da rezultatul corect putem reține un număr de zecimale semnificativ ale erorii și avem

$N = (0,04062 \pm 0,0006)mol = 0,04122mol$  sau  $0,04002mol$  ce corespunde unei precizii de 1,5% (=  $0,04122 / 0,0006$ ).

Pentru eroarea relativă avem

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

Deci

$$\delta f = \sum_{i=1}^n x_i^* \frac{\partial}{\partial x_i} \ln f(x^*) \delta x_i.$$

- *Eroare relativă* prin raportul:

$$\frac{\Delta f}{|f(x^0)|}$$

Calculul erori relative se poate face mai simplu folosind diferențiala funcției logaritmice

$$d \ln | f | = \frac{df}{f(x^0)}$$

**Exemplu:** Refaceți calculul de la exemplul anterior utilizând diferențiala logaritmică

$$\text{Soluție: } N = \frac{PV}{RT} \Rightarrow \ln(N) = \ln(P) + \ln(V) - \ln(R) - \ln(T) \Rightarrow$$

$$d(\ln N) = \frac{dN}{N} = \frac{dP}{P} + \frac{dV}{V} - \frac{dT}{T} \Rightarrow$$

$$\frac{\Delta N}{N} = \left| \frac{1}{P} \right| \Delta P + \left| \frac{1}{V} \right| \Delta V + \left| -\frac{1}{T} \right| \Delta T = 0,01 + 0,005 + \frac{0,1}{300} \approx 0,015 = 1,5\%$$

Se găsește același rezultat.

De o mare importanță practică este și problema inversă: cu ce precizie trebuie approximate datele pentru ca rezultatul să aibă o precizie dată? Adică, dându-se  $\varepsilon > 0$ , cât trebuie să fie  $\Delta x_i$  sau  $\delta x_i$ ,  $i = \overline{1, n}$  astfel încât  $\Delta f$  sau  $\delta f < \varepsilon$ ? O metodă de rezolvare se bazează pe *principiul efectelor egale*: se presupune că toți termenii care intervin în (3.3.1) sau (3.3.2) au același efect, adică

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

Se obține

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

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

# Reprezentarea numerelor în virgulă mobilă

## 1. Introducere

Un număr real  $x$  poate fi reprezentat în baza  $\beta$  (trebuie să fie pară) sub următoarea formă:

$$x = \pm d_0.d_1d_2 \dots d_{p-1} \times \beta^e, \quad 0 \leq d_i < \beta$$

unde cifrele  $d_i$  formează *semnificantul* sau *mantisa*,  $p$  este *precizia*, iar *exponentul*  $e_{min} \leq e \leq e_{max}$ .

Valoarea lui  $x$  este

Pentru a avea o reprezentare unică, numerele se normalizează astfel încât  $d_0 \neq 0$ .

Atunci este necesar ca pentru reprezentarea lui *zero* sa se adopte convenția

$$0 = 1.0 \times \beta^{e_{min}-1}$$

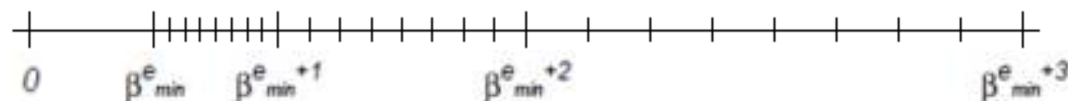


Fig. Reprezentarea numerelor normalizate în virgulă flotantă

Fiecare interval de forma  $[\beta^e, \beta^{e+1})$  din  $\mathbb{R}$  conține  $\beta^p$  numere în virgulă flotantă (numărul posibil de semnificanți).

Se poate observa din secvențele Matlab de mai jos densitatea numerelor în aceste subintervale:

```
>> eps(0)
ans =
    4.9407e-324
```

```
>> eps(2)
ans =
    4.4409e-016
```

```
>> eps(8)
ans =
    1.7764e-015
```

```
>> eps(1)
ans =
    2.2204e-016
```

```
>> eps(4)
ans =
    8.8818e-016
```

```
>>eps(1024)
ans =
    2.2737e-013
```

În intervalul  $(0, \beta^{e_{min}})$  care este gol se introduc numerele denormalizate, numere cu semnificantul de forma  $0.d_1d_2 \dots d_{p-1}$  și exponentul  $\beta^{e_{min}-1}$ .

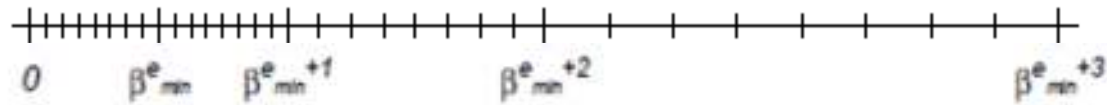


Fig. Reprezentarea numerelor denormalizate în virgulă flotantă

Multimea numerelor reprezentate în virgulă flotantă este o submulțime a numerelor raționale și se notează:

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

Spunem că avem depășire superioară sau inferioară dacă

$$|x| > \beta \times \beta^{e_{max}}, \quad \text{sau} \quad |x| < 1.0 \times \beta^{e_{min}}$$

Se definesc operațiile obișnuite pe această mulțime, dar trebuie să ținem cont ca:

$$\begin{aligned}(x \oplus y) \oplus z &\neq x \oplus (y \oplus z) & (x \otimes y) \otimes z &\neq x \otimes (y \otimes z) \\ (x \oplus y) \otimes z &\neq x \otimes z \oplus y \otimes z.\end{aligned}$$

De exemplu în Matlab avem:

```
>> (0.1+0.1)-0.2
ans =
     0
```

```
>> (0.1+0.1+0.1)-0.3
ans =
 5.5511e-017
```

Pentru măsurarea erorii de reprezentare, în afară de eroarea relativă, se folosește *ulps* – *units in the last place* (unități în ultima poziție). Dacă numărul  $z$  se reprezintă prin  $d_0.d_1d_2 \dots d_{p-1} \times \beta^e$ , atunci eroarea este

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



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  $\underbrace{0.0 \dots 0}_p \beta' \times \beta^e$ , cu  $\beta' = \frac{\beta}{2}$ . Valoarea  $\text{eps} = \frac{\beta}{2}\beta^{-p}$  se numește

*epsilon-ul mașinii.*

Rotunjirea implicită se face după regula cifrei pare: dacă  $x = d_0.d_1 \dots d_{p-1}d_p \dots$  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. Dacă notăm cu  $\text{fl}$  operația de rotunjire, operațiile aritmetice din  $\mathbb{F}$  se pot defini prin

$$x \odot y = \text{fl}(x \circ y).$$

Pentru operațiile în virgulă flotantă se poate folosi *axioma fundamentală a aritmeticii în virgulă flotantă*

$$\forall x, y \in \mathbb{F}, \exists \delta \text{ cu } |\delta| < \text{eps} \text{ astfel încât } x \odot y = (x \circ y)(1 + \delta).$$

ce ne spune căorice operație în virgulă flotantă este exactă până la o eroare relativă de cel mult eps.

În Matlab avem:

```
>> 1-3*(4/3-1)
```

```
ans =
```

```
2.220446049250313e-016
```

```
>> 1-3*(7/3-2)
```

```
ans =
```

```
-4.440892098500626e-016
```

Acest lucru se datorează reprezentării finite a unor numere infinite. Astfel  $4/3$  nu se poate reprezenta exact cu ajutorul unui număr finit de termeni binari.

$$\frac{4}{3} = \frac{1}{\frac{3}{4}} = \frac{1}{1 - \frac{1}{4}} = \sum_{k=0}^{\infty} \frac{1}{4}^k$$

$$\frac{4}{3} = 1 + \frac{1}{2^2} + \frac{1}{2^4} + \frac{1}{2^6} + \dots$$

$$\frac{4}{3} = 1.010101010101 \dots$$

## 2. Anularea

Fie  $x$  și  $y$  calculați cu erorile lor relative:

$$x \approx x(1 + \delta_x) \text{ și } y \approx y(1 + \delta_y)$$

Atunci:

$$\delta_{xy} = \delta_x + \delta_y,$$

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

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

Singura operație critică din punct de vedere al erorii este scăderea a două cantități apropiate  $x \approx y$ , caz în care  $\delta_{x-y} \rightarrow \infty$ . Acest fenomen se numește anulare și este reprezentat grafic în figura 3.3. Aici  $b, b', b''$  sunt cifre binare acceptabile, iar  $g$ -urile reprezintă cifre binare contaminate de eroare (gunoaie – garbage digits). De notat că, gunoi - gunoi = gunoi, dar mai important, normalizarea mută prima cifră contaminată de pe poziția a 12-a pe poziția a treia.

$x$	=	1	0	1	1	0	0	1	0	1	$b$	$b$	$g$	$g$	$g$	$g$	$e$
$y$	=	1	0	1	1	0	0	1	0	1	$b'$	$b'$	$g$	$g$	$g$	$g$	$e$
$x-y$	=	0	0	0	0	0	0	0	0	0	$b''$	$b''$	$g$	$g$	$g$	$g$	$e$
	=	$b''$	$b''$	$g$	$g$	$g$	$g$	?	?	?	?	?	?	?	?	?	$e-9$

Anularea este de două tipuri: *benignă*, când se scad două cantități exacte și *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. Vom da în continuare câteva exemple de anulări catastrofale și modul de transformare a lor .

## Exemple:

(R. Trîmbițaș, 2005, *Analiza numerică. O introducere bazată pe MATLAB*, Presa Universitară Clujeană)

**Exemplul 3.4.1.** 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  $b \gg a$ . ◇

**Exemplul 3.4.2.** Dacă anularea apare într-o expresie cu radicali, se amplifică cu conjugata:

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

**Exemplul 3.4.3.** Diferența valorilor unei funcții pentru argumente apropiate se transformă folosind formula lui Taylor:

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

## 2. Standardul IEEE754

Standardul IEEE754 prevede ca  $\beta = 2$  (reprezentarea în baza 2)

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

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

În cazul IEEE 754,  $D = 127$ .

Deplasamentul  $D$  este necesar pentru reprezentarea exponentilor negativi.

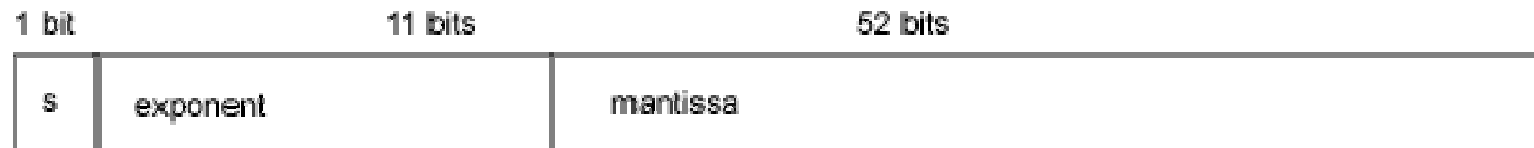
Deoarece în baza 2 singurele cifre sunt 0 și 1, prin normalizare  $d_0 = 1$ . Atunci acest bit nu se mai reprezintă și se mai câștigă un bit în reprezentarea mantisei:

$$(-1)^s 2^e (1 + f)$$

IEEE Floating Point Representation



IEEE Double Precision Floating Point Representation





În standardul IEEE 754 există următoarele cantități speciale:

Exponent	Semnificant	Ce reprezintă
$e = e_{min} - 1$	$f = 0$	$\pm 0$
$e = e_{min} - 1$	$f \neq 0$	$0.f \times 2^{e_{min}}$
$e_{min} \leq e \leq e_{max}$		$1.f \times 2^e$
$e = e_{max} + 1$	$f = 0$	$\pm \infty$
$e = e_{max} + 1$	$f \neq 0$	NaN

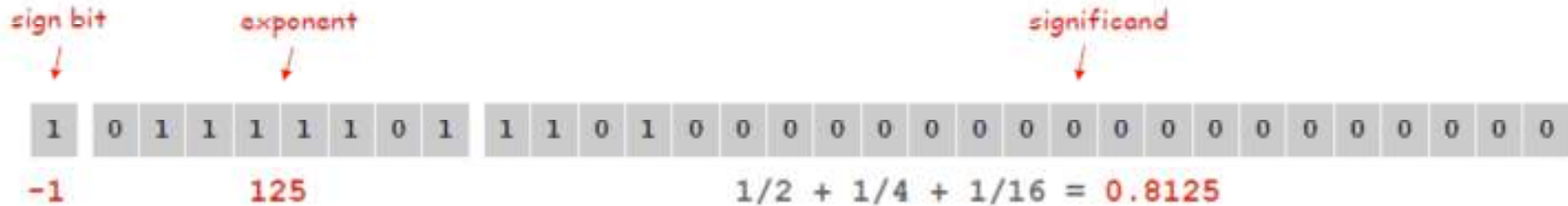
**NaN.** Avem de fapt o familie de valori NaN, operațiile ilegale sau nedeterminate conduc la NaN:  $\infty + (-\infty)$ ,  $0 \times \infty$ ,  $0/0$ ,  $\infty/\infty$ ,  $x \text{ REM } 0$ ,  $\infty \text{ REM } y$ ,  $\sqrt{x}$  pentru  $x < 0$ . Dacă un operand este NaN rezultatul va fi tot NaN.

**Infinit.**  $1/0 = \infty$ ,  $-1/0 = -\infty$ . 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  $\log x = \text{NaN}$  pentru  $x < 0$ . Fără 0 cu semn nu s-ar putea face distincție la logaritmi între un număr negativ care dă depășire superioară și 0.

Ex. Single precision representation of -0.453125.



$$-1 \times 2^{125 - 127} \times 1.8125 = -0.453125$$

The diagram shows the calculation of the floating point value. The sign bit is -1. The exponent is 125, and the bias is 127. The phantom bit is 1. The significand is 0.8125. The final result is -0.453125.

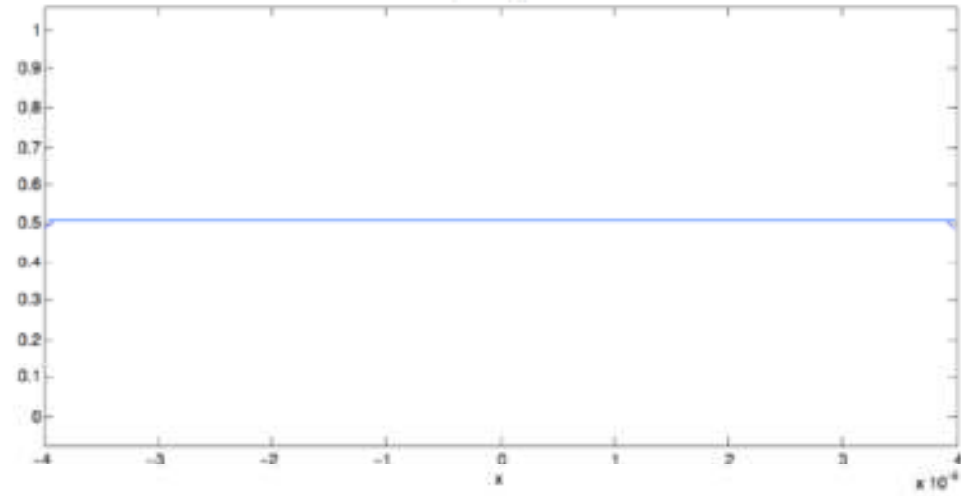
Exemplu: (<http://introc.cs.princeton.edu/java/lectures/9scientific.pdf>)

Reprezențați grafic funcția:

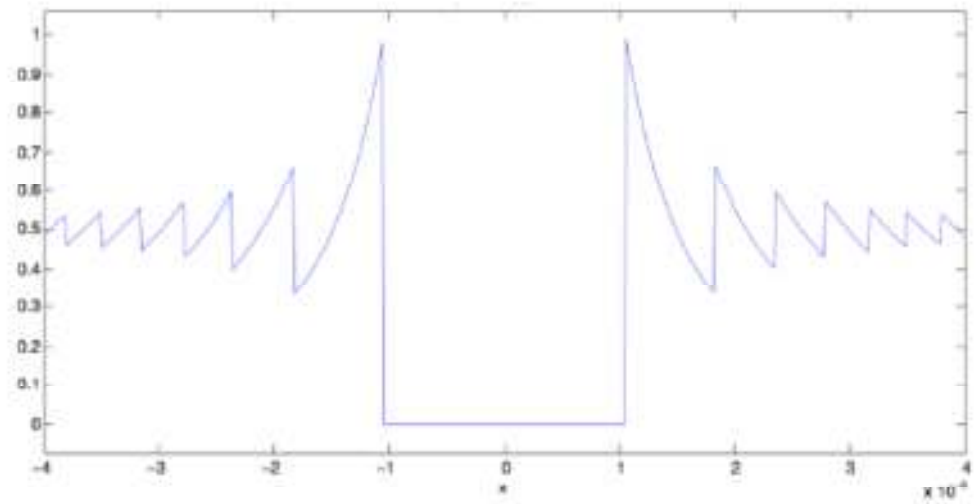
$$f(x) = \frac{1 - \cos x}{x^2}$$

$$-4 \cdot 10^{-8} \leq x \leq 4 \cdot 10^{-8}$$





Exact answer



IEEE 754 double precision answer

```
public static double fl(double x) {  
    return (1.0 - Math.cos(x)) / (x*x);  
}
```

Ex. Evaluate  $fl(x)$  for  $x = 1.1e-8$ .

- $Math.cos(x) = 0.9999999999999999888897769753748434595763683319091796875$ .  
    ← nearest floating point value agrees with exact answer to 16 decimal places.
- $(1.0 - Math.cos(x)) = 1.1102e-16$   
    ← inaccurate estimate of exact answer ( $6.05 \cdot 10^{-17}$ )
- $(1.0 - Math.cos(x)) / (x*x) = 0.9175$   
    ← 80% larger than exact answer (about 0.5)

**Catastrophic cancellation.** Devastating loss of precision when small numbers are computed from large numbers, which themselves are subject to roundoff error.

## Numerical Catastrophes

### Ariane 5 rocket. [June 4, 1996]

- 10 year, \$7 billion ESA project exploded after launch.
- 64-bit float converted to 16 bit signed int.
- Unanticipated overflow.



Copyright, Arianespace

### Vancouver stock exchange. [November, 1983]

- Index undervalued by 44%.
- Recalculated index after each trade by adding **change** in price.
- 22 months of accumulated truncation error.

### Patriot missile accident. [February 25, 1991]

- Failed to track scud; hit Army barracks, killed 28.
- Inaccuracy in measuring time in 1/20 of a second since using 24 bit binary floating point.



sau

<http://ta.twi.tudelft.nl/users/vuik/wi211/disasters.html>

MATLAB utilizează numere în virgulă flotantă dublă precizie, conform standardului IEEE. Nu se face distincție între numerele întregi sau reale. Comanda `format hex` este utilă la vizualizarea reprezentării în virgulă flotantă. De exemplu, reprezentările lui 1, -1, 0.1 și ale secțiunii de aur,  $\phi = (1 + \sqrt{5})/2$ , se obțin cu:

```
>> format hex
>> 1,-1
ans =
    3ff0000000000000
ans =
    bff0000000000000
>> 0.1
ans =
    3fb999999999999a
>> phi=(1+sqrt(5))/2
phi =
    3ff9e3779b97f4a8
```

Se consideră că fracția  $f$  satisface  $0 \leq f < 1$ , iar exponentul  $-1022 \leq e \leq 1023$ . Sistemul de numere în virgulă flotantă al MATLAB poate fi caracterizat de trei constante: `realmin`, `realmax` și `eps`. Constanta `realmin` reprezintă cel mai mic număr normalizat în virgulă flotantă. Orice cantitate mai mică decât ea este fie un număr denormalizat, fie dă depășire inferioară. Cel mai mare număr reprezentabil în virgulă flotantă se numește `realmax`. Orice cantitate mai mare decât el dă depășire superioară. Epsilon-ul mașinii este desemnat prin `eps`. Valorile acestor constante sunt

	binar	zecimal	hexazecimal
<code>eps</code>	$2^{-52}$	2.220446049250313e-016	3cb0000000000000
<code>realmin</code>	$2^{-1022}$	2.225073858507201e-308	0010000000000000
<code>realmax</code>	$(2-\text{eps}) \times 2^{1023}$	1.797693134862316e+308	7fefffffffffffffff

## Exemplu

```
>> format hex
>> 1.5
ans =
3ff8000000000000
```

**0011 1111 1111 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000**

Conversion Code - Chart																
DECIMAL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
BINARY	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

**0** semnul +  $\Rightarrow$  **s=0**

$e+1023 = 011\ 1111\ 1111 = 3*16^2 + 15*16 + 15 = 1023 \Rightarrow$  **e=0**

$f = 1000\ 0000\ 00\dots = 1*2^{-1} = 1/2 =$  **0,5**

Atunci

$$(-1)^0 * 2^0 * (1+f) = 1.5$$