

# Computació Numèrica

## Tema 1. Conceptes bàsics

M. Àngela Grau Gotés

Departament de Matemàtiques  
Universitat Politècnica de Catalunya · BarcelonaTech.

10 de febrer de 2020

“Donat el caràcter i la finalitat exclusivament docent i eminentment il·lustrativa de les explicacions a classe d'aquesta presentació, l'autor s'acull a l'article 32 de la Llei de propietat intel·lectual vigent respecte de l'ús parcial d'obres alienes com ara imatges, gràfics o altre material contingudes en les diferents diapositives”



© 2020 by M. Àngela Grau Gotés.

Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional.

# Índex

- 1 Introducció
- 2 Definicions d'error
  - Error absolut
  - Error relatiu
- 3 Classes d'errors
  - Errors d'arrodoniment
  - Errors de truncament
  - Propagació de l'error
- 4 Aritmètica de punt flotant
  - Representació de nombres
  - Nombres a l'ordinador
  - Aritmètica de Matlab
- 5 Algorismes
- 6 Exercicis

# Introducció

Durant els segles XX i XXI avançats models matemàtics s'han aplicat en diferents àrees de coneixement com l'enginyeria, la medicina, l'economia o les ciències socials. Sovint, les aplicacions generen problemes matemàtics que per la seva complexitat no poden ser resolts de manera exacta.

La **matemàtica computacional**, s'ocupa del disseny, anàlisi i implementació d'algorismes per obtenir solucions numèriques aproximades de models físics, químics, matemàtics, estadístics, ...

# Modelització

Davant d'un fet real, la modelització consisteix a construir un conjunt de fórmules i equacions que ens el representin de la manera més fidel possible, de manera que ens permeti fer prediccions correctes.

Els models resultants quasi mai no poden ser resolts per complet utilitzant mètodes (llapís i paper) d'anàlisi. La simulació en un ordinador ens permet interpretar els resultats i comparar-los amb les dades experimentals.

# Funció error

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

- S'anomena funció error, s'aplica si els resultats d'un conjunt de mesures es descriuen per una distribució normal de mitja zero i desviació estàndard  $\sigma$ , llavors  $\operatorname{erf}\left(\frac{\epsilon}{\sigma\sqrt{2}}\right)$  és la probabilitat que l'error en una de les mesures es trobi entre  $-\epsilon$  i  $\epsilon$ .
- Però la integral definida no es pot expressar per mitja de funcions elementals
- **Cal obtenir aproximacions numèriques!**

# Funció error

La sèrie de potències en un entorn de 0 és:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n!(2n+1)}$$

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \left( x - \frac{x^3}{3} + \frac{x^5}{10} - \frac{x^7}{42} + \frac{x^9}{216} - \dots \right)$$

Sèrie de convergència lenta.

# Algorismes

Un algorisme (matemàtic) és un procediment formal que descriu una seqüència ordenada i finita d'operacions a realitzar un nombre finit de vegades per tal d'obtenir la solució d'un problema.

Algorismes són com receptes amb els blocs bàsics de operacions **suma**, **resta**, **multiplicació**, i **divisió**, així com les estructures de programació: **for**, **while**, i **if** si es resol amb ajut d'ordinadors.



# Algorismes

En general, qualsevol algorisme que desenvolupem o del que fem ús ha de ser: exacte, estable, eficient i robust.

**Exacte** Què tan bo és l'algorisme d'aproximació de la quantitat a calcular (accuracy).

**Estable** La sortida de l'algorisme és sensible a petits canvis en les dades d'entrada (stability).

**Eficient** Quant costa (en nombre d'operacions) a obtenir una aproximació raonable (efficiency).

**Robust** Per a quants casos puc fer ús de l'algorisme (robustness).

En alguns casos també importa la memòria necessària (storage) i si és paral·lelitzable (parallelization).

En aquest curs treballarem amb dos tipus d'algorismes

**Directes** Obtenen la solució en un nombre finit de passos.

**Exemples** *Equacions de segon grau. El·liminació gaussiana.*

**Iteratius** Generen una seqüència de valors aproximats que convergeixen a la solució quan el nombre de passos tendeix a infinit.

**Exemple** *El mètode iteratiu següent convergeix a  $\sqrt{2}$ .*

$$x_k = \frac{1}{2} \left( x_{k-1} + \frac{2}{x_{k-1}} \right) \quad k \geq 1 \text{ i } x_0 = 3.$$

Els sis primers iterats de  $x_k = \frac{1}{2} \left( x_{k-1} + \frac{2}{x_{k-1}} \right)$  són:

$n$	$x_n$	$ x_n - \sqrt{2} $
0	3.0000000000000000	$1.58578643762690 \times 10^0$
1	1.8333333333333333	$4.19119770960238 \times 10^{-1}$
2	1.4621212121212121	$4.79076497481170 \times 10^{-2}$
3	1.414998429894803	$7.84867521707922 \times 10^{-4}$
4	1.414213780047198	$2.17674102520604 \times 10^{-7}$
5	1.414213562373112	$1.66533453693773 \times 10^{-14}$
6	1.414213562373095	$2.22044604925031 \times 10^{-16}$

# Fonts d'error

Fixarem quatre grans fonts d'error, que poden influir en una aproximació "pobre" del fet observat:

- Error de modelització. Una elecció equivocada o inapropiada de model.
- Error de truncament/discretització. Aproximacions discretes i finites del model matemàtic (algorismes).
- Error experimental. Mesures incorrectes o dolentes.
  - ▶ Errors aleatoris. Les mesures.
  - ▶ Errors sistemàtics. Calibració incorrecte.
  - ▶ Errors aberrants. Per descomptat també hi ha el factor humà. Un error per descuit o ignorància.
- Error d'arrodoniment/de tall. Error acumulat a causa de l'execució del nostre model on les operacions descrites per l'algorisme es fan amb un nombre finit de dígit.

# Algorismes i error

Aquest curs principalment, ens centrarem en estudiar els errors en el procés de càlcul, l'algorisme emprat per a resoldre:

**De truncament** Convertir un procés infinit en finit.

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$$

**D'arrodoniment** Deguts a l'aritmètica de punt flotant de l'ordinador.

Menys importants que els de truncament, però poden esdevenir catastròfics.

# Algorismes i error

Nosaltres ens centrarem en estudiar els errors en el procés de càlcul, errors produïts en interrompre els processos infinits, errors d'arrodoniment i de truncament.

Per a això, farem el següent:

- Definir el que entenem per error.
- Analitzar els errors associats a l'aritmètica flotant.
- Presentar algorismes que minimitzen aquest error.

# 2

## Error absolut. Error relatiu

# Error absolut

Notem per  $x$ , el valor exacte i per  $\tilde{x}$ , un valor aproximat

## Definició

$$\Delta x = x - \tilde{x} \quad (1)$$

## Pràctica

$$\begin{aligned} x &= 1/3, & \tilde{x} &= 0.3333 \\ x &= \pi, & \tilde{x} &= 3.141 \end{aligned}$$

Un defecte que té aquesta definició és que no considera la magnitud del valor.



# Error relatiu

Notem per  $x$ , el valor exacte i per  $\tilde{x}$ , un valor aproximat

## Definició

$$\epsilon_x = \frac{\Delta x}{x} \quad (2)$$

## Pràctica

$$\begin{array}{ll} x = 1/3, & \tilde{x} = 0.3333 \\ x = \pi, & \tilde{x} = 3.141 \end{array}$$

*Error relatiu aproximat:*  $\epsilon_{\tilde{x}} = \frac{\Delta x}{\tilde{x}}$ .

*Error relatiu percentual:* l'error relatiu en tant per 100.

# Exacte i exactitud

## Definició: xifres decimals correctes (accurate)

Direm que  $\tilde{x}$  és una aproximació a  $x$  amb  $d$  xifres decimals correctes si  $d$  és el nombre natural més gran tal que

$$|x - \tilde{x}| < 0.5 \cdot 10^{-d} \quad (3)$$

## Definició: xifres significatives correctes (precision)

Direm que  $\tilde{x}$  és una aproximació a  $x$  amb  $t$  xifres significatives si  $t$  és el nombre natural més gran tal que

$$\frac{|x - \tilde{x}|}{|x|} < 0.5 \cdot 10^{-t} \quad (4)$$

Accuracy-Precision (D. W. Harder)

Whats the difference between precision and accuracy?

# Estimacions - Cotes error

Per a  $x = \pi$  i  $\tilde{x} = 3.141$ , tenim

$$\Delta x = 0.00059265 \dots \quad \epsilon_x = 0.000188647 \dots$$

$$|\Delta x| \leq 0.6 \cdot 10^{-3}, \quad x = 3.141 \pm 0.6 \cdot 10^{-3}.$$

$$|\epsilon_x| \leq 0.2 \cdot 10^{-3} = 0.02\%, \quad x = 3.141 \cdot (1 \pm 0.02\%).$$

# Estimacions (II)

**Exemple:** Sigui  $x = \sqrt{2} = 1.414213562\dots$  i  $\bar{x} = 1.414$ . Aleshores

$$e_a(\sqrt{2}) = -0.0002135\dots$$

i

$$\begin{aligned} e_r(\sqrt{2}) &= \frac{-0.0002135\dots}{\sqrt{2}} \simeq \frac{-0.0002135}{1.414} \\ &= -0.00015099\dots \end{aligned}$$

Per l'exemple anterior tenim que

$$\varepsilon_a = 0.00022, \quad \varepsilon_r = 0.00016$$

# Autoavaluació

**Exercici 1** Calculeu l'error absolut, l'error relatiu i l'error relatiu aproximat de les quantitats:

$$\begin{aligned}x &= 9234.567, & \tilde{x} &= 9234.564; \\x &= 0.634, & \tilde{x} &= 0.631.\end{aligned}$$

Què s'observa?

**Exercici 2** Calculeu l'error absolut, l'error relatiu, les xifres correctes de les quantitats:

$$\begin{aligned}x &= 1/3, & \tilde{x} &= 0.3333, \\x &= 1/3, & \tilde{x} &= 0.3334,\end{aligned}$$

# Autoavaluació

**Exercici 3** Calculeu les xifres significatives de les quantitats:

$$\begin{aligned}x &= 10000, & \tilde{x} &= 9998, \\x &= 10000, & \tilde{x} &= 9999.99998, \\x &= 0.0000025, & \tilde{x} &= 0.0000018,\end{aligned}$$

**Exercici 4** Calculeu l'error absolut i l'error relatiu de les aproximacions següents:

$$\begin{aligned}a) & \quad x = \pi, \quad \tilde{x} = 22/7 \\b) & \quad x = e, \quad \tilde{x} = 2.718 \\c) & \quad x = 10^\pi, \quad \tilde{x} = 1400 \\d) & \quad x = e^{10}, \quad \tilde{x} = 22000\end{aligned}$$

# 3.1

## Errors d'arrodoniment

# Errors d'arrodoniment

La representació decimal d'un **nombre real** es redueix per tal representar/usar els nombres reals a l'ordinador o en càlculs manuals.

La representació decimal d'un nombre és pot reduir per tall o per arrodoniment a un nombre finit de dígit.

Exemple.

Representar  $\frac{2}{3}$  per una expressió decimal de 5 dígit.



# Arrodonir nombres decimals

Sigui  $x$  qualsevol nombre decimal positiu de la forma

$$x = 0.d_1d_2 \dots d_{n-1}d_nd_{n+1} \dots d_m$$

llavors  $\widetilde{r}_x$ , l'arrodoniment de  $x$  a  $n$  xifres decimals ( $n < m$ ) depèn del valor del dígit  $n + 1$ .

$$\widetilde{r}_x = 0.d_1d_2 \dots d_{n-1}d, \quad (5)$$

$$d = \begin{cases} d_n & \text{si } d_{n+1} \in \{0, 1, 2, 3, 4\}, \\ d_n + 1 & \text{si } d_{n+1} \in \{5, 6, 7, 8, 9\}. \end{cases} \quad (6)$$

# Estimació error arrodoniment

## Teorema

*Si el nombre  $x$  s'arrodoneix, i  $\widetilde{r}_x$  és el seu valor arrodonit a  $n$  dígit, aleshores la fita de l'error absolut és*

$$|x - \widetilde{r}_x| \leq \frac{1}{2} \cdot 10^{-n}. \quad (7)$$

## Exercici.

Representeu els nombres .1735499 .99995000 i .4321609 amb quatre dígit per arrodoniment.

# Tallar nombres decimals

La aproximació per tall del nombre  $x$  a  $n$  dígit ( $n < m$ ) és el nombre  $\widehat{t}_x$  obtingut en descartat tots els dígit posteriors al dígit  $n$ .

$$x = 0.d_1d_2 \dots d_n \dots d_m \xrightarrow[n \text{ dígit}]{\text{tallar}} \widehat{t}_x = 0.d_1d_2 \dots d_n, \quad (8)$$

# Estimació error tallar

## Teorema

Si el nombre  $x$  es talla, i  $\widehat{t}_x$  és el seu valor aproximat a  $n$  dígits, aleshores la fita de l'error absolut és

$$|x - \widehat{t}_x| \leq 10^{-n}. \quad (9)$$

## Exercici.

Representeu els nombres .1735499 .99995000 i .4321609 amb quatre dígits per tall.

# Precisió implícita

Per escriure una mesura com un nombre decimal (o binari o  $\dots$ ), hi ha un nivell de precisió implícit, si no es diu el contrari 0.5 unitats en l'última posició escrita. Altrament, és convenient escriure la dada amb l'error màxim explícitament.

- Si escrivim 23.4567 hem d'entendre  $23.4567 \pm 0.00005$
- Altrament escriuriem la fita de l'error,  $23.4567 \pm 0.0012$
- La precisió implícita de Matlab sempre és la mateixa.

**Quan es mostri un valor aproximat, cal que la precisió reflecteixi l'exactitud (error absolut/relatiu).**

# 3.2

## Errors de truncament

# Errors de truncament (I)

Els errors de truncament sorgeixen en el cas d'aproximar un procés infinit per un de finit.

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$$

Els errors de discretització o en substituir una expressió contínua per una discreta. El procés numèric és generalment una aproximació del model matemàtic obtingut com a funció d'un paràmetre de discretització, que serà notat per  $h$ , i suposarem positiu. Si, quan  $h$  tendeix a 0, el procés numèric torna la solució del model matemàtic, direm que el procés numèric és convergent.

# Errors de truncament (II)

truncament procès infinit

$$S = \sum_{n=1}^{+\infty} (-1)^n a_n \approx S_k = \sum_{n=1}^k (-1)^n a_n$$

Leibniz va obtenir la següent sèrie matemàtica (1682):

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \dots = \frac{\pi}{4}$$



# Errors de discretització (I)

El procés numèric és generalment una aproximació del model matemàtic obtingut com a funció d'un paràmetre de discretització, que serà notat per  $h$ , i suposarem positiu. Si, quan  $h$  tendeix a 0, el procés numèric torna la solució del model matemàtic, direm que el procés numèric és convergent. A més, si l'error (absolut o relatiu) es pot fitar en funció de  $h$ , de la forma

$$e_d \leq Ch^p, \quad C > 0, \quad p > 0 \iff e_d = \mathcal{O}(h^p)$$

es diu que *el mètode és convergent d'ordre  $p$*  i escrivim

# Errors de discretització (II)

Per definició la derivada d'una funció  $f(x)$  és:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

llavors podem fer les aproximacions numèriques ( $h > 0$ ):

## discretització

$$f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} + \frac{f''(\xi)}{2}h, \quad \text{fórmula de Taylor}$$

llavors

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h} \quad e_d = \mathcal{O}(h)$$

# 3.3

## Propagació de l'error

# Propagació de l'error: funcions

Si  $f : \mathbb{R} \rightarrow \mathbb{R}$  és una funció derivable,  $x$  un nombre real,  $\tilde{x}$  una aproximació de  $x$  amb fita d'error  $\epsilon$ ,  $x = \tilde{x} \pm \epsilon$ , el teorema del valor mig diu

$$|f(x) - f(\tilde{x})| = |f'(\xi)| |x - \tilde{x}|, \quad |\tilde{x} - x| \leq \epsilon.$$

Així, una manera de valorar l'efecte que tenen els errors en les dades d'entrada en el càlcul de  $y = f(x)$  seria:

## Fórmula de la propagació de l'error absolut

$$|\Delta f| \approx |f'(\tilde{x})| \epsilon. \quad (10)$$

# Propagació de l'error: funcions

Si  $g : \mathbb{R}^2 \rightarrow \mathbb{R}$  és una funció diferenciable i  $x_1 = \tilde{x}_1 \pm \epsilon_1$  i  $x_2 = \tilde{x}_2 \pm \epsilon_2$ , llavors

## FGPE en dues variables

$$|\Delta g| \approx \left| \frac{\partial g(\tilde{x}_1, \tilde{x}_2)}{\partial x_1} \right| |\epsilon_1| + \left| \frac{\partial g(\tilde{x}_1, \tilde{x}_2)}{\partial x_2} \right| |\epsilon_2|. \quad (11)$$

Si fem ús d'una funció real de dues variables serà possible fitar l'error propagat per les operacions elementals.

# Propagació de l'error: operacions

Suma,  $g(x_1, x_2) = x_1 + x_2$

$$x_1 + x_2 = (\tilde{x}_1 + \tilde{x}_2) \pm (\delta x_1 + \delta x_2) \quad (12)$$

Resta,  $g(x_1, x_2) = x_1 - x_2$

$$x_1 - x_2 = (\tilde{x}_1 - \tilde{x}_2) \pm (\delta x_1 + \delta x_2) \quad (13)$$

Les cotes dels errors absoluts es sumen en les operacions de sumar i restar nombres reals.

Les cotes dels errors relatius es sumen en les operacions de multiplicar i dividir nombres reals.

Producte,  $g(x_1, x_2) = x_1 \cdot x_2$

$$|\delta g| \approx |\tilde{x}_2| |\delta x_1| + |\tilde{x}_1| |\delta x_2|, \quad \left| \frac{\delta g}{g} \right| \approx \left| \frac{\delta x_1}{\tilde{x}_1} \right| + \left| \frac{\delta x_2}{\tilde{x}_2} \right|. \quad (14)$$

Divisió,  $g(x_1, x_2) = x_1/x_2$

$$|\delta g| \approx \left| \frac{1}{\tilde{x}_2} \right| |\delta x_1| + \left| \frac{\tilde{x}_1}{\tilde{x}_2^2} \right| |\delta x_2|, \quad \left| \frac{\delta g}{g} \right| \approx \left| \frac{\delta x_1}{\tilde{x}_1} \right| + \left| \frac{\delta x_2}{\tilde{x}_2} \right|. \quad (15)$$

Sigui  $g : \mathcal{D} \rightarrow \mathbb{R}$ ,  $\mathcal{D}$  una regió de  $\mathbb{R}^n$ ,  
 $g$  una funció diferenciable en un entorn del vector  $\tilde{x}$   
 $x = \tilde{x} \pm \Delta x$ , amb  $\Delta x = (\Delta x_1, \dots, \Delta x_n)^t$ .

### Error absolut propagat

$$|\Delta y| = |y - \tilde{y}| \approx \sum_{i=1}^n \left| \frac{\partial g(\tilde{x})}{\partial x_i} \right| \Delta x_i. \quad (16)$$

La fórmula (16) de l'error absolut propagat s'obté de la fórmula de Taylor aplicada a  $y = g(x)$  i  $\tilde{y} = g(\tilde{x})$ .



# Nombres de condició

## Error relatiu propagat

$$\left| \frac{\Delta y}{\tilde{y}} \right| \approx \sum_{i=1}^n \left| \frac{\tilde{x}_i}{g(\tilde{x})} \frac{\partial g(\tilde{x})}{\partial x_i} \right| \left| \frac{\Delta x_i}{\tilde{x}_i} \right|. \quad (17)$$

L'expressió (17) s'obté dividint per  $\tilde{y}$  i multiplicant i dividint per  $\tilde{x}_i$ .  
Els  $n$  valors

$$\left| \frac{\tilde{x}_i}{g(\tilde{x})} \frac{\partial g(\tilde{x})}{\partial x_i} \right| \quad (18)$$

s'anomenen números de condició o factors de propagació. Aquests donen una mesura de quan un problema és mal condicionat.

# 4

## Aritmètica de punt flotant

What Every Computer Scientist Should Know About Floating-Point Arithmetic

# Representació de nombres (punt fix)

$$x = \pm d_1 d_2 d_3 \dots d_n \cdot d_{n+1} d_{n+2} \dots d_{n+m}$$

Representa el nombre real  $x$  expressat en base 10, on cada  $d_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .

## Exemples

$$2345 = (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (5 \times 10^0) = 2000 + 300 + 40 + 5.$$

$$45.67 = (4 \times 10^1) + (5 \times 10^0) + (6 \times 10^{-1}) + (7 \times 10^{-2}) = 40 + 5 + 0.6 + 0.07,$$

L'aritmètica es realitza desplaçant el punt decimal.

# Representació de nombres (punt flotant)

$$x = \pm 0.d_1 d_2 d_3 \dots d_n \times 10^e \quad d_1 \neq 0.$$

Representa el nombre real  $x$  expressat en **base 10** i **precisió  $n$**  en notació de punt flotant **normalitzada**, on l'exponent  $e$  és un nombre enter i cada dígit  $d_i$  un enter entre 0 i 9.

## Exemples

Els nombre  $-15.24$  en punt flotant és  $-0.1524 \cdot 10^2$ ;

el nombre  $0.000617$  en punt flotant és  $0.617 \cdot 10^{-3}$ ;

el nombre  $4.274952 \cdot 10^{15}$  és  $0.4274952 \cdot 10^{16}$ ;

i  $-6543219$  en punt flotant és  $-0.6543219 \cdot 10^7$ .

# Nombres binaris

El sistema binari, és com el decimal però només fan ús dels dígit 0 i 1.

## Exemples

$$\begin{aligned}(101.1101)_{(2)} &= (1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4})_{10} \\ &= (4 + 1 + 1/2 + 1/4 + 1/32)_{10} = (5.78125)_{10}\end{aligned}$$

$$97 = (1000011)_{dos}, \text{ restesdividirper } 2$$

$$0.7 = (0.\overline{10110})_{dos}, \text{ partenterademultiplicarper } 2$$

# 4.2

## Nombres a l'ordinador

# Norma 754 — 1985

L'any 1985, l'*Institute for Electrical and Electronic Engineers (IEEE)* va publicar l'informe

*Binary Floating Point Arithmetic Standard 754 — 1985*,  
en el que s'especifiquen normes per representar nombres en punt flotant amb precisió simple, doble i extensa. L'informe va ser revisat i actualitzat l'any 2008, *IEEE Std 754-2008*.

Avui en dia, quasi tots els fabricants d'ordinadors han acceptat aquesta norma; per tant l'ordinador emmagatzema no el nombre real  $x$  si no una aproximació binària (octal o hexadecimal) en punt flotant a  $x$ .

# El conjunt $F(\beta, t, L, U)$

El conjunt de nombres en punt flotant representables a l'ordinador el designarem per  $F(\beta, t, L, U)$ , on  $\beta$  representa la **base**,  $t$  la **precisió** (o nombre de dígit representats o significatius) i el interval  $[L, U]$  és el rang de l'exponent  $e$ .

Per a tot nombre real  $x$  expressat en el conjunt  $F(\beta, t, L, U)$  existeixen  $t$  xifres i un exponent  $e$  tal que

$$fl(x) = \pm \left( \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \dots + \frac{d_t}{\beta^t} \right) \cdot \beta^e,$$

amb dígit  $d_i \in \mathbb{N}$  tal que  $0 \leq d_i < \beta$  per a tot  $i = 1 \div t$ ; i exponent  $L \leq e \leq U$ . Hi pot haver  $U - L + 1$  exponents diferents.



# El conjunt $F(\beta, t, L, U)$

La quantitat

$$\pm \left( \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \dots + \frac{d_t}{\beta^t} \right)$$

s'anomena fracció o part fraccionària del nombre  $x$ .

Si exigim  $d_1 \neq 0$  per a  $x \neq 0$ , resulta que  $\beta^{-1} \leq |f| < 1$  i es diu que la representació és normalitzada.

La quantitat  $m = d_1 d_2 d_3 \dots d_t$  s'anomena **mantissa**.

Hi pot haver  $\beta^t$  mantisses diferents.

# Precisió de l'ordinador/software

La precisió d'un ordinador dependrà del fabricant i del tipus de variable que es defineixi; la unitat d'informació ve donada pel nombre de dígits binaris o longitud de la paraula (word): *1 byte = 8 bits*, *1 word = 2 bytes(PC)*, *4 bytes(VAX)*, *8 bytes(CRAY)*.

La longitud de les paraules imposa una restricció sobre la precisió amb la un ordinador pot representar nombres reals. Això vol dir que l'ordinador emmagatzema no el nombre real  $x$  si no una aproximació binària a aquest nombre, usualment es designa per  $fl(x)$ .

# 32 bits

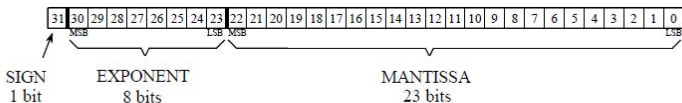
La representació en punt flotant amb simple precisió ocupa 1 paraula; té 32 dígitos binaris, assignats de la manera següent,

signe del nombre real $x$	1 bit
exponent, (enter)	7 bits
mantissa, (real)	23 bits

Un nombre real  $x$  en simple precisió es representa en 32 bits binari per

$$fl(x) = (-1)^s \times M \times 2^{c-127}.$$

amb  $M = 1.m_{22}m_{21} \dots m_1m_0 = 1 + f$



#### Example 1

0	00000111	110000000000000000000000	
↓	↓	↓	
+	7	0.75	$+ 1.75 \times 2^{(7-127)} = + 1.316554 \times 10^{-36}$

#### Example 2

1	10000001	011000000000000000000000	
↓	↓	↓	
-	129	0.375	$- 1.375 \times 2^{(129-127)} = - 5.500000$

FIGURE 4-2

Single precision floating point storage format. The 32 bits are broken into three separate parts, the sign bit, the exponent and the mantissa. Equations 4-1 and 4-2 shows how the represented number is found from these three parts. MSB and LSB refer to "most significant bit" and "least significant bit," respectively.

# 64 bits

La representació en punt flotant amb doble precisió

$$fl(x) = (-1)^s \times M \times 2^{c-1023}$$

ocupa 2 paraules; té 64 dígit binaris, assignats de la manera següent

$s$ signe del nombre real $x$	1 bit
$c$ exponent, (enter)	11 bits
$m$ mantissa, (real)	52 bits

Si es demana  $M = 1.m_{51}m_{50} \dots m_1m_0 = 1 + f$  augmenta en un bit la precisió.

Té entre 15 i 16 dígit decimals de precisió.

# 64 bits

La condició de 11 bits per l'exponent significa que  $|e| \leq 2^{11} - 1 = 2047$ , restant 1023 obtenim el rang  $(-1023, 1024)$  per a l'exponent. Si s'acorda la representació en que el primer dígit binari de la mantissa sigui 1,  $M = 1.m_{51}m_{50} \dots m_1m_0 = 1 + f$ , podem tindre matises de 53 dígits binaris, que es corresponen almenys 15 xifres decimals de precisió.

El nombre positiu més petit és  $s = 0$ ,  $c = 1$  i  $f = 0$  que es correspon al nombre decimal  $2^{-1022} \cdot (1 + 0) \approx 0.2225 \cdot 10^{-307}$ . El nombre positiu més gran és  $s = 0$ ,  $c = 2046$  i  $f = 1 - 2^{-52}$  que es correspon al nombre decimal  $2^{1023} \cdot (1 + f) \approx 0.17977 \cdot 10^{309}$ .

# Èpsilon de la màquina

Si l'ordinador té l'aritmètica  $F(\beta, t, L, U)$ , llavors

$$fl(x) = x(1 + \delta) \quad |\delta| \leq \epsilon_M = \frac{1}{2}\beta^{1-t}.$$

La **precisió** d'una aritmètica de coma flotant es caracteritza per l'**èpsilon de la màquina**, no és el nombre més petit representable, però dona una mesura relativa de fins a on dos nombres molt pròxims seran diferents.

El seu valor es correspon a la meitat distància entre 1 i el següent nombre en punt flotant.

En Matlab és  $\epsilon = 2.2204e - 016$ .

# Aritmètica a $F(\beta, t, L, U)$

Les operacions aritmètiques en coma flotant són

$$\begin{array}{ll} x + y = \longleftrightarrow & x \oplus y = fl(fl(x) + fl(y)) \\ x - y = \longleftrightarrow & x \ominus y = fl(fl(x) - fl(y)) \\ x \times y = \longleftrightarrow & x \otimes y = fl(fl(x) \times fl(y)) \\ x \div y = \longleftrightarrow & x \oslash y = fl(fl(x) \div fl(y)) \end{array}$$

En totes les operacions s'ha de complir

$$x \circledast y = fl(x \circledast y)(1 + \epsilon_M).$$



# Aritmètica a $F(\beta, t, L, U)$

Imagem un ordinador  $F(10, 5, 0, 127)$ , i els nombres  $x = .31426 \cdot 10^3$  i  $y = .92577 \cdot 10^5$ .

$x \times y =$	$.2909324802 \cdot 10^8$	$x \otimes y =$	$.29093 \cdot 10^8$
$x + y =$	$.9289126 \cdot 10^5$	$x \oplus y =$	$.92891 \cdot 10^5$
$x - y =$	$-.92262740 \cdot 10^5$	$x \ominus y =$	$-.92263 \cdot 10^5$
$x \div y =$	$.3394579647 \cdot 10^{-2}$	$x \oslash y =$	$.33946 \cdot 10^{-2}$






En aquests resultats l'error relatiu és de  $8.5 \cdot 10^{-6}$ ,  $2.3 \cdot 10^{-6}$ ,  $2.8 \cdot 10^{-6}$ ,  $6.0 \cdot 10^{-6}$ , respectivament, tots per sota de  $10^{-5}$ .

# 4.3

## Aritmètica de Matlab

# Aritmètica de Matlab

Cal llegir els documents:

-  [Aritmètica en punt flotant by Cleve Moler](#)  
Fall96Cleve.pdf
-  [Cleve's Corner: Cleve Moler on Mathematics and Computing, Floating Point Arithmetic Before IEEE 754](#)
-  [Cleve's Corner: Cleve Moler on Mathematics and Computing, Floating Point Numbers](#)
-  [Cleve's Corner: Cleve Moler on Mathematics and Computing, Floating Point Denormals, Insignificant But Controversial](#)
-  [MathWorks Documentation Center, floating-point-numbers.html](#)

# 5

## Estabilitat numèrica i problemes ben condicionats

Some disasters attributable to bad numerical computing

# Estabilitat numèrica

Un algorisme el classifiquem com **numèricament estable** si un error, no creix **gaire** en el procés de càlcul.

L'estabilitat numèrica es veu afectada pel nombre de xifres significatives, poques xifres o la pèrdua en passos intermitjos del càlcul disminueix la fiabilitat dels resultats obtinguts.

# Anàlisi de l'error

Si  $f$  representa a l'algoritme *real* i  $f^*$  a l'algoritme *computacional*,  $x$  a la variable *real* i  $x^*$  a la variable *computacional*, aleshores l'error en el resultat final es pot definir com:

$$\begin{aligned} |f(x) - f^*(x^*)| \leq & \underbrace{|f(x) - f(x^*)|}_{\text{condició}} + \\ & \underbrace{|f(x^*) - f^*(x)|}_{\text{estabilitat}} + \\ & \underbrace{|f^*(x) - f^*(x^*)|}_{\text{truncament}} \end{aligned}$$

# Algorismes amb cancel·lació

La **pèrdua de xifres significatives** per cancel·lació, que es produeix en restar dos nombres molt propers. La situació es pot resumir en

$$g(x + \delta) - g(x) \quad \text{amb } |\delta| \ll 1 ; \quad (19)$$

## Exemple.

Les solucions de  $x^2 - 18x + 1 = 0$  són  $x_{1,2} = 9 \pm \sqrt{80}$ .

Si  $\sqrt{80} = 8.9443 \pm 0.5 \cdot 10^{-4}$  llavors  $x_1 = 17.9443 \pm 0.5 \cdot 10^{-4}$ , té 6 xifres significatives, mentre que  $x_2 = 0.0557 \pm 0.5 \cdot 10^{-4}$ , només en té 3.

# Inestabilitat numèrica

Sense rigor, diem que un procés numèric és inestable quan els petits errors que es produeixen en un dels seus estadis s'agranden en etapes posteriors, fins a tal punt que no podem fiar-nos del càlcul global.

## Exemple.

Per calcular les integrals  $I_n = \int_0^1 x^n e^{x-1} dx$ ,  $n \geq 1$ , disposem de dos mètodes iteratius diferents:

$$\text{a) } I_{n-1} = \frac{1 - I_n}{n}, \quad n \geq 2 \quad \text{on } I_0 = 0,$$

$$\text{b) } I_n = 1 - nI_{n-1}, \quad n \geq 2 \quad \text{on } I_1 = 1/e.$$



# Sensibles a les condicions inicials

Molts problemes són especialment sensibles a les dades inicials, independentment dels errors d'arrodoniment i de l'algorisme emprat.

**Exemple.** Polinomi de Wilkinson

Sigui  $p(x) = (x - 1)(x - 2)(x - 3)\dots(x - 10)$ , el polinomi amb arrels els deu primers nombres naturals, definim el polinomi

$q(x) = p(x) + \frac{1}{2^{13}} x^9$ , modificant lleugerament el coeficient de  $x^9$

respecte de  $p(x)$ . Com haurien de ser les arrels del polinomi  $q(x)$ ? Calculeu-les.

# Sensibles a les condicions inicials

Són problemes on la solució depèn de manera molt sensible de les dades. Si petites variacions de les dades provoquen grans variacions en la solució, es diu que el problema està mal condicionat.

## Exemple

Resoleu els sistemes

$$\left\{ \begin{array}{rcl} 2x & - & 4y = 1 \\ -2.998x & + & 6.001y = 2 \end{array} \right. \quad \left\{ \begin{array}{rcl} 2x & - & 4y = 1 \\ -3x & + & 6.001y = 2 \end{array} \right.$$

# Evitar la propagació dels errors

Per tal de reduir o evitar la propagació dels errors es recomana, minimitzar el nombre d'operacions, reordenar les operacions i replantejar el problema en altres termes.

## equació de segon grau

Resoldre l'equació  $x^2 + 62.10x + 1 = 0$  treballant amb quatre dígitos i arrodonint

## regla de horner

Avaluar el polinomi  $P(x) = x^3 - 6.1x^2 + 3.2x + 1.5$  per  $x = 4.71$  fent ús d'una aritmètica de tres dígitos.

# Exercicis

# Operacions en punt flotant

En una aritmètica de cinc dígit, representeu els nombres  $x = 1/3$  i  $y = 5/7$  i calculeu:

a)  $x \times y$  i  $x \otimes y$ .

b)  $x + y$  i  $x \oplus y$ .

c)  $x - y$  i  $x \ominus y$ .

d)  $x \div y$  i  $x \oslash y$ .

Comproveu que l'error es manté per sota de  $0.5 \cdot 10^{-4}$ .

# Respostes: Operacions en punt flotant

Si  $fl(1/3) = 0.33333$  i  $fl(5/7) = 0.71428$  llavors:

a)  $x \times y = 5/21$  i  $x \otimes y = fl(0.2380909524) = 0.23809$

els errors són  $\delta = 0.524 \cdot 10^{-4}$  i  $\epsilon = 0.220 \cdot 10^{-4}$ .

b)  $x + y = 22/21$  i  $x \oplus y = fl(1.04761) = 1.04761$

els errors són  $\delta = 0.190 \cdot 10^{-4}$  i  $\epsilon = 0.182 \cdot 10^{-4}$ .

c)  $x - y = -8/21$  i  $x \ominus y = fl(-0.38095) = -0.38095$

els errors són  $\delta = 0.238 \cdot 10^{-4}$  i  $\epsilon = 0.625 \cdot 10^{-5}$ .

d)  $x \div y = 7/15$  i  $x \oslash y = fl(0.466665733325867) = 0.46666$

els errors són  $\delta = 0.667 \cdot 10^{-4}$  i  $\epsilon = 0.143 \cdot 10^{-4}$ .

# Problemes amb operacions

En una aritmètica de cinc dígit, representeu els nombres  $y = 5/7$ ,  $u = 0.714251$ ,  $v = 98765.9$  i  $w = 0.111111 \cdot 10^{-4}$  i calculeu:

a)  $y \ominus u$ . (restar dues quantitats molt properes)

b)  $(y \ominus u) \oslash w$ . (dividir per una quantitat petita)

c)  $(y \ominus u) \otimes v$ . (multiplicar per una quantitat gran)

d)  $u \oplus v$ .

e)  $y \ominus w$ .

Comproveu que l'error NO es manté per sota de  $0.5 \cdot 10^{-4}$ .

# Respostes: Problemes amb operacions

$$\begin{aligned}y &= 5/7 & \Rightarrow & fl(y) = 0.71428 \cdot 10^0 \\u &= 0.714251 & \Rightarrow & fl(u) = 0.71425 \cdot 10^0 \\v &= 98765.9 & \Rightarrow & fl(v) = 0.98765 \cdot 10^5 \\w &= 0.111111 \cdot 10^{-4} & \Rightarrow & fl(w) = 0.11111 \cdot 10^{-4}\end{aligned}$$

<i>Oper.</i>	<i>Error abs.</i>	<i>Error rel.</i>
$y \ominus u$	$0.472 \cdot 10^{-5}$	0.136
$(y \ominus u) \oslash w$	0.425	0.136
$(y \ominus u) \otimes v$	0.466	0.136
$u \oplus v$	$0.162 \cdot 10^1$	$0.164 \cdot 10^{-4}$
$y \ominus w$	0.779	$0.122 \cdot 10^{-4}$



# Autoavaluació

**Exercici 1** Realitzeu les operacions aritmètiques:

$$\frac{4}{5} + \frac{1}{3}; \quad \frac{4}{5} \cdot \frac{1}{3}; \quad \left(\frac{1}{3} - \frac{3}{11}\right) + \frac{3}{20}; \quad \left(\frac{1}{3} + \frac{3}{11}\right) - \frac{3}{20};$$

a) Fent ús d'una aritmètica de tres xifres i tallant els nombres.

b) Fent ús d'una aritmètica de tres xifres i arrodonint els nombres.

c) Calculeu els errors relatius dels apartats a) i b).

Exercici 2 Calculeu:  $\sum_{k=1}^6 \frac{1}{3^k}$  i  $\sum_{k=1}^6 \frac{1}{3^{(7-k)}}$

- a) Fent ús de l'aritmètica de tres xifres arrodonint.
- b) Fent ús de l'aritmètica de quatre xifres arrodonint.
- c) Per què donen diferent? Calculeu en cada cas l'error relatiu percentual.