POLYTECH INFORMATIQUE

3ÈME ANNÉE

Alexandra Bac

NUMERICAL METHODS

COURS 1
INTRODUCTION, NOMBRES ET MACHINES

MÉTHODES NUMÉRIQUES

↓

Calculer avec un ordinateur

MÉTHODES NUMÉRIQUES

↓

Calculer avec un ordinateur

↙ ↘

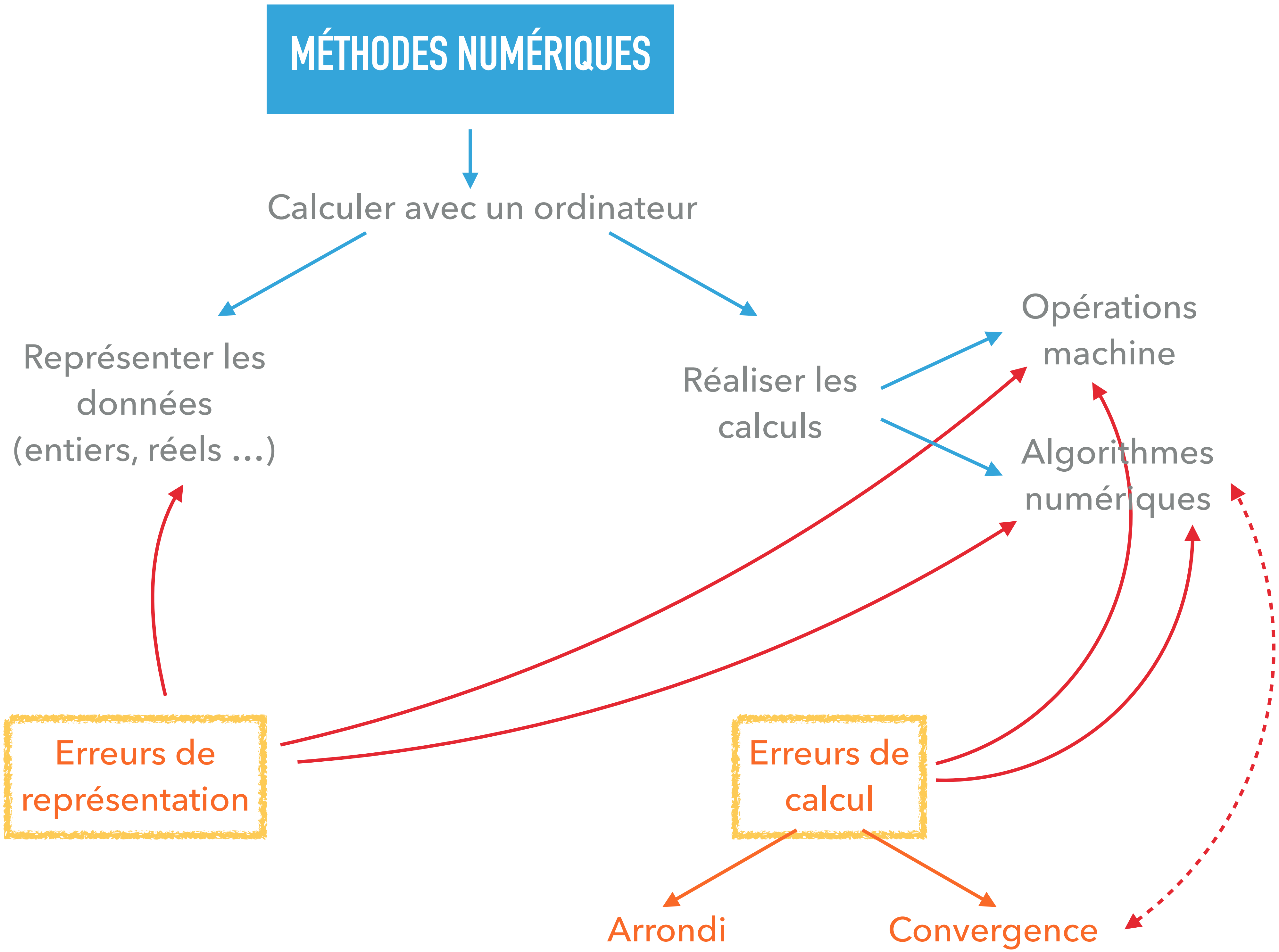
Représenter les données (entiers, réels …)

Réaliser les calculs

MÉTHODES NUMÉRIQUES

Calculer avec un ordinateur

- Représenter les données (entiers, réels …)
- Réaliser les calculs
  - Opérations machine
  - Algorithmes numériques



π, $\sqrt{2}$ ...

0.1 → cannot be rep. exactly on a computer ...

ANALYSE NUMÉRIQUE

Représentation

COURS 1

Erreurs

Algorithmes

Complexité / Convergence

Stabilité des solutions

Algorithmie numérique

$f(x) = 0$

COURS 1
- Zéros de fonctions

COURS 2 (2)
- Stabilité systèmes linéaires
- Algorithmique des systèmes linéaires

COURS 3
- Valeurs singulières
- Analyse en composantes principales

COURS 4
- Intégration numérique
- Dérivation numérique

(2) Gauss alg. ✓

Stability → conditioning number of a syst.

REPRÉSENTATION DES ENTIERS

PROBLÈME DE LA REPRÉSENTATION

Représentation en base $b \quad (b \in \mathbb{N}, b > 0)$

$$n = \sum_{i=1}^{N} c_i \cdot b^i \quad \text{avec } c_i < b$$

Représenté par

$$(c_0, \ldots, c_N)$$

$b = 2$

$$n = \sum_{i=1}^{N} c_i \cdot b^i \quad \text{avec } c_i \in \{0,1\}$$

bits

Représenté par

$$(c_0, \ldots, c_N)$$

N bits

PROBLÈME DE LA REPRÉSENTATION

Représentation en base b $\quad (b \in \mathbb{N}, b > 0)$

$$n = \sum_{i=1}^{N} c_i \cdot b^i \quad \text{avec } c_i < b$$

Représenté par

$$(c_0, \ldots, c_N)$$

$b = 2$

$$n = \sum_{i=1}^{N} c_i \cdot b^i \quad \text{avec } c_i \in \{0,1\}$$

bits

Représenté par

$$(c_0, \ldots, c_N)$$

N bits

Ordinateur → objets finis

integer ~ 32 bits / 4 bytes

Entiers négatifs ?

PROBLÈME DE LA REPRÉSENTATION

$$n = \sum_{i=1}^{N} c_i \cdot b^i \quad \text{avec } c_i \in \{0,1\}$$

ENTIERS POSITIFS

$$N \leftrightarrow 0 \leqslant n \leqslant 2^N - 1$$

- N=8 (unsigned char)
  $2^n - 1 = 255$
- N=16 (unsigned short)
  $2^n - 1 = 65535$
- N=32 (unsigned int)
  $2^n - 1 = 4.294.967.295$
- N=64 (unsigned long int)
  $2^n - 1 = 1{,}844 \cdot 10^{19}$

ENTIERS RELATIFS

1 bit utilisé pour coder le signe

$$N \leftrightarrow -2^{N-1} \leqslant n \leqslant 2^{N-1} - 1$$

- N=8 (char)
  $-128 \leqslant n \leqslant 127$
- N=16 (short)
  $-32768 \leqslant n \leqslant 32767$
- N=32 (int)
  $-2.147.483.648 \leqslant n \leqslant 2.147.483.647$
- N=64 (long int)
  $-9{,}2 \cdot 10^{18} \leqslant n \leqslant 9{,}2 \cdot 10^{18}$

$n \longrightarrow$ representation over N bits ?

yes

if

$n \in$ range given on prev. slide

$n = (c_0 \dots c_N)$

NO

$300 \longrightarrow$ cannot be represented with $N = 8$

can be represented with $N = 16$

PROBLÈME DE LA REPRÉSENTATION DES ENTIERS RELATIFS

$$n = \sum_{i=1}^{N} c_i \cdot b^i \quad \text{avec } c_i \in \{0,1\}$$

Codage du signe

CODAGE NAÏF PAR UN BIT

$$\boxed{c_{N-1}} c_{N-2} \ldots c_1 c_0 \rightarrow n = (-1)^{c_{N-1}} \left(\sum_{i=0}^{n-2} c_i \cdot 2^i \right)$$

Bit de signe

-11 sur 8 bits :

1 | 0 | 0 | 0 | 1 | 0 | 1 | 1

COMPLEMENT A UN

Le plus utilisé actuellement

$$c_{N-1} c_{N-2} \ldots c_1 c_0 \rightarrow n = -c_{N-1} \cdot 2^{N-1} + \sum_{i=0}^{n-2} c_i \cdot 2^i$$

-11 sur 8 bits :

$$-11 = \boxed{-128} + \boxed{117}$$

1 | 1 | 1 | 1 | 0 | 1 | 0 | 1

PROBLÈME DE LA REPRÉSENTATION DES ENTIERS RELATIFS

$$n = \sum_{i=1}^{N} c_i \cdot b^i \quad \text{avec } c_i \in \{0,1\}$$

Codage du signe

CODAGE NAÏF PAR UN BIT

- 0 a deux représentations
  00000000 / 10000000
- Addition → cas à traiter
  -11 + (-5)
  -11 + 5

COMPLEMENT A UN

Le plus utilisé actuellement

- 0 a une seule représentation
  00000000
- L'addition reste valable

$$\begin{array}{rr} -11 : & 11110101 \\ -5 : & 11111011 \\ \hline + & 1\boxed{11110000} \end{array} \longrightarrow 112 \text{ (déc)}$$

Résultat : -128 + 112 = -16 (déc)

PROBLÈME DE LA REPRÉSENTATION DES ENTIERS RELATIFS

$$n = \sum_{i=1}^{N} c_i \cdot b^i \quad \text{avec } c_i \in \{0,1\}$$

Codage du signe

CODAGE NAÏF PAR UN BIT

- 0 a deux représentations
  00000000 / 10000000
- Addition → cas à traiter
  -11 + (-5)
  -11 + 5

COMPLEMENT A UN

Le plus utilisé actuellement

- 0 a une seule représentation
  00000000
- L'addition reste valable

$$\begin{array}{rr} -11: & 11110101 \\ 5: & 00000101 \\ \hline + & 1\boxed{1111010} \end{array} \longrightarrow 122 \text{ (déc)}$$

Résultat : -128 + 122 = -6 (déc)

REPRÉSENTATION DES FLOTTANTS …

Mériterait un cours entier … cf TD …

UN MOT SUR LES ERREURS

EVALUATION DE L'ERREUR COMMISE

$\boxed{\text{Erreur commise entre } x \text{ et } x' \text{ ?}}$

$x = 1{,}2957$
$x' = 1{,}2956$
$|x - x'| = 10^{-4}$

$x = 0{,}0017$
$x' = 0{,}0016$
$|x - x'| = 10^{-4}$

$x = 0{,}0007$
$x' = 0{,}0006$
$|x - x'| = 10^{-4}$

Même erreur ?

EVALUATION DE L'ERREUR COMMISE

Erreur commise entre x et x' ?

$x = 1{,}2957$

$x = 1{,}2956$

$|x - x'| = 10^{-4}$

$7 \cdot 10^{-3}\,\%$

$x = 0{,}0017$

$x = 0{,}0016$

$|x - x'| = 10^{-4}$

Même erreur ?

$5{,}9\,\%$

$x = 0{,}0007$

$x = 0{,}0006$

$|x - x'| = 10^{-4}$

$14{,}3\,\%$

EVALUATION DE L'ERREUR COMMISE

Erreur commise entre x et x' ?

ERREUR ABSOLUE

$$\boxed{|x - x'|}$$

- Naturel sur les petits nombres
  $|x| < 1$
- Peut être inatteignable sur de grands nombres

ERREUR RELATIVE

$$\boxed{\frac{|x - x'|}{|x|}}$$

- Naturel sur les grands nombres
  $|x| \gg 1$
- L'erreur dépend de l'ordre de grandeur de x

Compute solution x^*

converging alg.

$u_0 \quad u_1 \quad \dots\dots u_n \longrightarrow x^*$

$\lim_{n \to \infty}$

$u_N \rightsquigarrow$ STOP

$$\boxed{\|u_n - u_{n-1}\| < \varepsilon \quad / \quad \frac{\|u_n - u_{n-1}\|}{\|u_n\|} < \varepsilon}$$

x^* up to a given precision

error → absolute

error → relative

!

Matlab — Octave — Linux
Windows) free

Linear syst. Matlab $A - 10^4 \times 10^4$ ~ few minutes

$AX = B$ Octave → longer ++++

$$A = \left[\begin{array}{c|c} B & I \\ \hline 0 & \cdots \end{array}\right]$$

Math. operations :

$A \times B$ — algebra

alg. prod

$*$

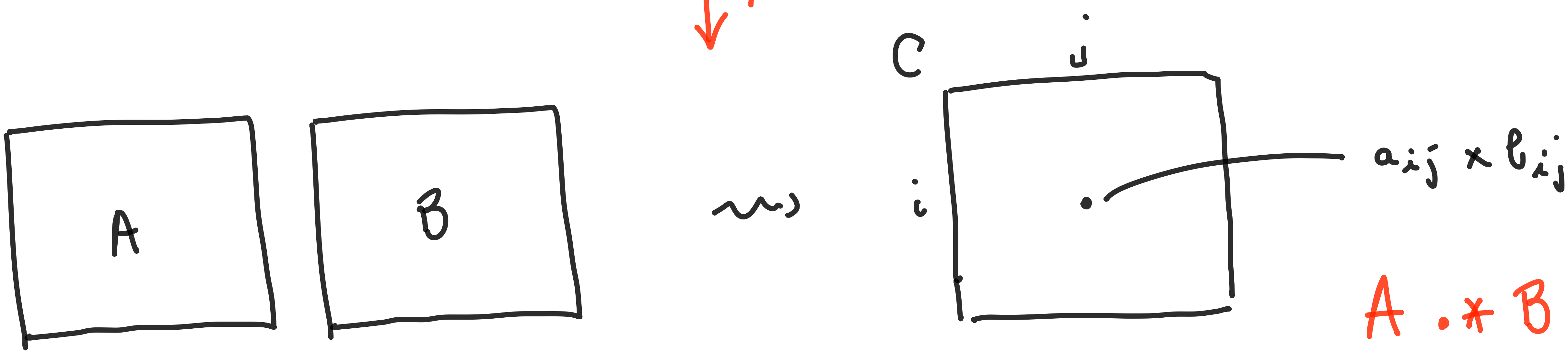
$$(A \times B)_{ij} = \sum_k A_{ik} \times B_{kj}$$

Alg.

$$A^n = \underbrace{A \times \dots \times A}_{n\times} \quad \rightsquigarrow \quad \wedge$$

$$A\wedge 3 = A \times A \times A$$

$\neq$



algorithmic

$A \mathbin{.*} B$

//

for i ...
 for j ...
  $c_{ij} = a_{ij} \times b_{ij}$

RAND

uniform → [0, 1]

normal → probability