MN

1

Présentation générale.

Intérêts des maths numériques

① Programmation.

arithmétique flottante (erreurs d'arrondis).
instabilité numérique ; complexité d'algorithme.

② Simulations numériques

Modélisation, approximation, simulations numériques, visualisation.
CAO, simulation 3D (effets spéciaux, images synthèse, jeux vidéo).

③ Traitement des images

Compression d'images (SVD, FFT, ondelettes)
Débruitage, reconnaissance de forme.

④ Analyse de données (statistiques appliquées).

Moindres carrés, ACP, Classification

Plan du cours

I] Arithmétique flottante
II] Système linéaire
- Décomposition LU
- Conditionnement (stabilité).

III] Interpolation
IV] Moindres carrés (SVD)
V] ACP (Analyse en composantes principales).

Chapitre 1
Arithmétique flottante.

I) Représentation d'un nombre en virgule flottante.

① Introduction.

On ne peut pas représenter exactement un nombre dans un ordinateur $\frac{1}{3} \neq 0,3333$

La représentation naturelle compte :
- la mantisse
- un exposant.

Exemple : $\underbrace{2,475678}_{\text{mantisse}} \cdot \underbrace{10^{38}}_{\text{exposant}}$

Cette représentation dépend de la base choisie

Définition et convention :

On dit que la représentation est normalisée si la mantisse est de la forme $c_0, c_1 c_2 c_3 \ldots$ avec $c_0 \neq 0$.

Exemple : $9,4751$ normalisée $\neq$ $0,0763 \cdot 10^3$ non normalisée.

Ecriture d'un nombre réel en base β.

$x = (c_0, c_1 c_2 \ldots) \cdot \beta^e$ $\qquad x = \left(\sum_{i=0}^{+\infty} c_i \beta^{-i} \right) \beta^e$.

$x = \left(c_0 \cdot \beta^0 + c_1 \cdot \beta^{-1} + \ldots \right) \cdot \beta^e$

② Représentation finie d'un nombre flottant.

On restreint le nombre de digits pour la mantisse et l'exposant

$\Rightarrow$ On définit un ensemble fini de nombres

$\mathbb{F}(\beta, p, e_{min}, e_{max})$.

où β : base de la représentation.

p : nombre de chiffres dans la mantisse

e_{min}, e_{max} : exposant minimum, maximum.

Tout nombre de $\mathbb{F}(\beta, p, e_{min}, e_{max})$ s'écrit :

$$x = \pm \underbrace{(c_0, c_1 c_2 \dots c_{p-1})}_{p \text{ chiffres.}} \cdot \beta^e$$

où $e \in [e_{min}, e_{max}]$.

et $0 \leq c_i \leq \beta - 1$.

ou bien encore :

$$x = \pm \left(\sum_{i=0}^{p-1} c_i \, \beta^{-i} \right) \beta^e.$$

et représentation normalisée si $c_0 \neq 0$.

Remarque : des nombres de $\mathbb{F}$ ne sont pas répartis uniformément.

Notation : On notera M le plus grand nombre positif de $\mathbb{F}(\beta, p, e_{min}, e_{max})$ et m le plus petit nombre positif normalisé.

Détermination de M et de m

$$M = \left(\sum_{i=0}^{p-1} (\beta - 1)\, \beta^{-i} \right) \beta^{e_{max}}.$$

$$= (\beta - 1)\, \frac{1 - \beta^{-p}}{1 - \beta^{-1}} \cdot \beta^{e_{max}}.$$

Soit $\boxed{M = (1 - \beta^{-p})\, \beta^{e_{max}+1}}$.

$m = (1,0\ldots0)\, \beta^{e_{min}}$

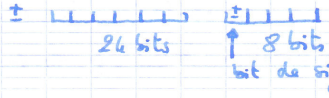
soit $\boxed{m = \beta^{e_{min}}}$.

Exemples : ($\beta = 2$)

a) Format simple précision (32 bits ou 4 octets).

mantisse : 24 bits (3 octets)

exposant : 8 bits (1 octet).



$e_{max} = 2^0 + 2^1 + \ldots + 2^6 = 2^7 - 1 = 128 - 1 = 127.$

$e_{min} = -127.$

Mais le zéro est représenté avec des 0 dans la mantisse et un exposant $e_{min} - 1$.

Or on veut $e_{min} - 1 = -127$, donc on choisit $\boxed{e_{min} = -126}$.

D'où $\mathbb{F}(2, 24, -126, 127)$.

On obtient donc $M = 3{,}4028 \cdot 10^{38}$

$m = 1{,}17549 \cdot 10^{-38}$.

b) Format double précision : (64 bits ou 8 octets)

mantisse 53 bits
exposant 11 bits
$\Rightarrow \mathbb{F}(2, 53, -1022, 1023)$

$M = 1,798.10^{308}$.
$m = 2,225.10^{-308}$.

Notations :

On notera $+\inf$, $-\inf$
Nan : des nombres spéciaux
↳ Not a number.

③ Approximation d'un nombre réel par un nombre flottant (représentation finie).

Un nombre flottant désigne un nombre de $\mathbb{F}(\beta, p, e_{min}, e_{max})$.
A chaque $x \in \mathbb{R}$, on associe la représentation $fl(x)$.
(fl pour "float") dans la représentation $\mathbb{F}(\beta, p, e_{min}, e_{max})$.

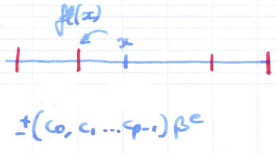
Construction de $fl(x)$:

(i) Overflow : Si $|x| > M$ alors

$$fl(x) = \begin{cases} +\inf & \text{si } x > M \\ -\inf & \text{si } x < -M \end{cases}$$

(ii) Underflow : Si $|x| < m$ alors $fl(x) = 0$.

(iii) Si $m \leq |x| \leq M$, alors $fl(x)$ est le flottant le plus proche de x (perfect rounding)



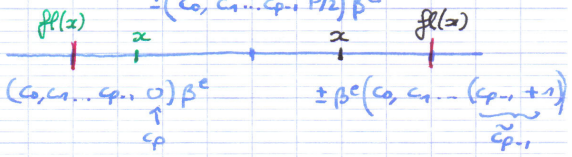
$\pm(c_0, c_1 \ldots c_{p-1})\beta^e$

MN
4

Si $x = \pm \left(\sum_{i=0}^{+\infty} c_i \beta^{-i} \right) \beta^e = \pm (c_0, c_1 \dots c_{p-1} c_p \dots) \beta^e$.

$\pm (c_0, c_1 \dots c_{p-1} \beta/2) \beta^e$

ⓐ Si $c_p < \beta/2$.



(*) $fl(x) = \pm \left(\sum_{i=0}^{p-1} c_i \beta^{-i} \right) \beta^e$.

ⓑ Si $c_p \geq \beta/2$ avec au moins un indice $j > p$ tq $c_j \neq 0$.

(**) Alors $fl(x) = \pm \left(\sum_{i=0}^{p-2} c_i \beta^{-i} + (c_{p-1} + 1) \beta^{-(p-1)} \right) \beta^e$

ⓒ Si $c_p = \beta/2$ avec $c_j = 0$ pour $j > p$.

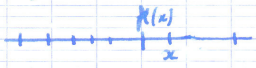
Alors il y a deux façons d'arrondir :

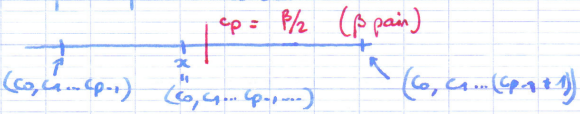
1: $fl(x)$ donné par (*) [cas a]

2: On arrondit pour que le dernier chiffre de la représentation soit pair (norme IEEE).

Soit $x \in \mathbb{R}$, $x = (c_0, c_1 \dots) \beta^e = \left(\sum_{i=0}^{+\infty} c_i \beta^{-i} \right) \beta^e$.

On approche x par $fl(x) \in \mathbb{F}(\beta, p, e_{min}, e_{max})$ de sorte que $fl(x)$ soit le nombre flottant le plus proche de x





Cas $m \leq |x| \leq M$

(i) si $c_p < \frac{\beta}{2}$ $\quad fl(x) = (c_0, c_1 \dots c_{p-1}) \beta^e \quad (*)$

(ii) si $c_p \geq \beta/2$ avec $c_j \neq 0$ pour un $j > p$

alors $fl(x) = (c_0, c_1 \dots c_{p-2} \tilde{c}_{p-1}) \beta^e$ (**) avec $\tilde{c}_{p-1} = c_{p-1} + 1$

(iii) si $c_p = \beta/2$ et $c_j = 0 \quad \forall j > p$

on a 2 possibilités: ⓐ arrondi classique : $fl(x)$ est donné par (*)

ⓑ Norme IEEE : on arrondit pour que le dernier chiffre de la représentation soit pair :

si c_{p-1} est pair alors $fl(x)$ est donné par (*)

si c_{p-1} est impair alors $fl(x)$ est donné par (**)

Exemple : $\mathbb{F}(10, 8, -38, 37) \sim \mathbb{F}(2, 24, -126, 127)$ [simple précision]

$fl(\underbrace{9,9999999}_{c_0 \; c_{p-1} \; c_p}) = 10$

$fl(\underbrace{1,5555555}_{c_p = \beta/2 = 5}000) = \begin{cases} 1,5555556 & \text{arrondi classique} \\ 1,5555556 & \text{norme IEEE.} \end{cases}$

$fl(\underbrace{2,5444444}_{c_{p-1} \; c_p}500) = \begin{cases} 2,5444445 & \text{arrondi classique} \\ 2,5444444 & \text{norme IEEE.} \end{cases}$

$fl(2,54444451) = 2,5444445.$

④ Epsilon machine : erreur entre un réel et sa représentation flottante

Soit $x \in \mathbb{R}$, $m \leq |x| \leq M$ $\quad x = \pm \left(\sum_{i=0}^{+\infty} c_i \beta^{-i} \right) \beta^e$.

et

$fl(x) \in \mathbb{F}(\beta, p, e_{min}, e_{max})$ le plus proche de x

la différence entre deux membres flottants consécutifs dans $[\beta^e ; \beta^{e+1}]$ est β^{1-p+e}

Preuve : $x = (c_0, c_1 \dots c_{p-1}) \beta^e$ $\quad$ donc $x < y$

$y = (\tilde{c}_0, \tilde{c}_1 \dots \tilde{c}_{p-1}) \beta^e$

$y - x = (c_0, 0 \dots 01) \beta^e = \beta^{-(p-1)} \beta^e = \beta^{1-p+e}$

On a donc $\quad |x - fl(x)| \leq \frac{1}{2} \beta^{1-p+e}$.

d'erreur relative $\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2} \beta^{1-p+e} \times \frac{1}{\beta^e}$ car $|x| \geq \beta^e$.

d'où

$$e_r = \frac{|x - fl(x)|}{|x|} \leq \frac{1}{2} \beta^{1-p}$$

borne indépendante de e c'est-à-dire de x. elle dépend uniquement du système de représentation flottant.

Epsilon machine : $\varepsilon_m = \frac{1}{2} \beta^{1-p}$.

d'où $fl(x) = x(1+\varepsilon)$ avec $|\varepsilon| \leq \varepsilon_m$.

Exemple: $\mathbb{F}(2, 24, -126, 127)$ simple précision.

$\varepsilon_m = \frac{1}{2} . 2^{1-24} \simeq 6 . 10^{-8}$

$\mathbb{F}(2, 53, -1022, 1023)$ double précision.

$\varepsilon_m = \frac{1}{2} 2^{1-53} \simeq 10^{-16}$.

Remarque: d'epsilon machine ε_m est le plus petit nombre positif tel que $fl(1+\varepsilon_m) > 1$.

II] La norme IEEE (IEEE 754).

Caractéristiques :

(i) 2 systèmes de flottants : $\mathbb{F}(2, 24, -126, 127)$
$\mathbb{F}(2, 53, -1022, 1023)$.

(ii) Nombres spéciaux : $\pm$ inf , NaN.
Si $|x| > M$ alors $fl(x) = \pm$ inf.
(dépassement de capacité : overflow).
On note $\oplus$; $\ominus$; $\otimes$; $\oslash$ les opérateurs effectués en machine.

On a $+\text{inf} \oplus x = +\text{inf}$ si $x \neq -\text{inf}$ et $x \neq$ NaN.
$\pm\text{inf} \otimes x = \pm\text{inf}$ si $x \neq 0$ et $x \neq$ NaN.
$\pm\text{inf} \oslash x = \pm\text{inf}$ si $x \neq \text{inf}$ et $x \neq$ NaN.
$\pm\text{inf} \oslash \pm\text{inf} =$ NaN
$\text{inf} \ominus \text{inf} =$ NaN $\quad 0 \oslash 0 =$ NaN.
$\text{inf} \otimes 0 =$ NaN
$x \oslash \text{inf} = 0$ si $x \neq \text{inf}$ et $x \neq$ NaN.

$p \leftarrow 0$, $\varepsilon \leftarrow 1$
tant que $(1 + 0 + \varepsilon > 1 - 0)$
$\varepsilon \leftarrow \varepsilon / 2$
$p \leftarrow p + 1$.

(iii) Modes d'arrondis vus précédemment.

(iv) Les 4 opérations élémentaires doivent être réalisées avec la précision la plus grande possible :
Soient u et v deux nombres flottants tels que $|u \cdot v| \in [m, M]$ où "$\cdot$" désigne une des 4 opérations élémentaires $(+, -, \times, /)$

On a : $u \odot v = fl(u \cdot v)$ où $\odot$ désigne l'opération machine.

MN6

III Conséquences sur les calculs.

① Introduction.

On se place dans la norme IEEE
On a $u \odot v = fl(u \cdot v) = (u \cdot v)(1+\epsilon)$ et $|\epsilon| \leq \varepsilon_m$.

⚠ L'addition n'est plus associative : $(u \oplus v) \oplus w \neq u \oplus (v \oplus w)$
et il peut y avoir une amplification importante des erreurs.

⚠ La multiplication et la division ne sont plus associatives mais il n'y a pas d'amplification d'erreurs.
Si on fait n multiplications, l'erreur relative sera de l'ordre de $n\varepsilon_m$.

② Soustraction de deux nombres proches.

La soustraction de deux flottants proches est exacte :

Proposition : En arithmétique IEEE, si u et v sont deux flottants

tels que $\frac{v}{2} \leq u \leq 2v$
Alors :

$$u - v = u \ominus v.$$

⚠ Mais l'opération $\ominus$ est instable.

Lorsqu'on effectue la soustraction de deux nombres flottants proches et de même signe qui proviennent d'un calcul, les erreurs sont amplifiées.

Soient x et y deux flottants de X et Y réels.

$x = X + \delta x$

$y = Y + \delta y$

$x \ominus y = fl(x-y) = (x-y)(1+\varepsilon)$ avec $|\varepsilon| \leq \varepsilon_m$.

Erreur relative : $$\frac{|x \ominus y - (X-Y)|}{|X-Y|} = \frac{|\delta x - \delta y|}{|X-Y|}$$

Or $|X-Y|$ peut être très petit, et donc

$\frac{|\delta x - \delta y|}{|X-Y|}$ très grand.

Conclusion : Les opérations de multiplication, division et addition de nombres de même signe sont des opérations stables. En revanche, la soustraction de nombres de même signe et proches est instable et doit nécessiter un traitement particulier (réécriture des formules).