

Module Mathématiques Numériques

Telecom Nancy

Bruno Pinçon (I.E.C.L.) bruno.pincon@univ-lorraine.fr

2018-2019

Module MN

Enseignements : 11 CM + 9 TD + 3 TP

Evaluation :

- 3 tests de 20 minutes (t_1 , t_2 , t_3 noté sur 20)
- 1 examen de 2 heures (e , noté sur 20)

Note finale :

$$NF = \max\left\{\frac{2}{3} \times e + \frac{1}{3}CC, \frac{1}{3} \times e + \frac{2}{3}CC\right\} \text{ avec } CC := \frac{t_1 + t_2 + t_3}{3}$$

Plan du cours

Chapitres :

- ① Arithmétique flottante (les pseudos nombres réels de l'ordinateur)
- ② Résolution de systèmes linéaires (factorisations), Normes vectorielles et matricielles
- ③ Interpolation
- ④ Approximation par moindres carrés
- ⑤ Notions sur la classification (analyse de données)

Chapitre 1 : Arithmétique flottante

- 1 Introduction
- 2 Définition des ensembles de flottants
- 3 Approximation d'un nombre réel par un nombre flottant
- 4 Règles des 4 opérations usuelles
- 5 Flottants utilisés sur ordinateurs
- 6 Un exemple de calcul d'erreur

Introduction I

La représentation en *virgule flottante* (appelée aussi *notation scientifique* et *floating point representation* en anglais) est naturelle pour écrire un nombre grand ou petit en valeur absolue. Exemples :

$$A = 6,02252 \cdot 10^{23} \text{ (le nombre d'Avogadro)}$$

$$h = 6,625 \cdot 10^{-34} \text{ [joule seconde] (la constante de Planck)}$$

Elle comporte deux parties :

- la mantisse (ici 6,02252 et 6,625) ;
- la partie exposant (23 et -34), le 10 est ici obligatoire car c'est la base choisie pour écrire la mantisse et l'exposant.

Introduction II

La représentation est *normalisée* si ^a : $c_0, c_1 c_2 c_3 \dots$ avec $c_0 \neq 0$

a. Autre convention $0, c_1 c_2 c_3 \dots$ avec $c_1 \neq 0$

Remarques :

- 1 Tout nombre réel (sauf zéro) peut s'écrire avec cette notation en virgule flottante normalisée avec en général un nombre de chiffres infini pour la mantisse (résultat provenant de la densité de \mathbb{Q} dans \mathbb{R}) ; on peut aussi remarquer que certains nombres peuvent s'écrire de deux façons, par exemple $0,999\dots = 1$.
- 2 Un changement de base peut introduire quelques bizarreries dans l'écriture d'un nombre alors que celle-ci est anodine dans la base initiale, par exemple (cf TD) :

$$(0,2)_{10} = (0,0011\,0011\,0011\,\dots)_2.$$

Introduction III

- 3 La mantisse d'un nombre rationnel écrit en virgule flottante est soit finie, soit infinie mais dans ce cas avec un pattern périodique de chiffres, cf exemple précédent ou encore :

$$\frac{1}{3} = (0, 33333.....)_{10}, \quad \frac{1}{7} = (0, \underline{142857} 142857 142857.....)_{10}$$

Définition des ensembles de flottants I

Dans un ordinateur on est obligé de restreindre le nombre de chiffres pour les mantisses et de limiter l'étendue des exposants ! On obtient alors des ensembles notés $\mathbb{F}(\beta, p, e_{min}, e_{max})$ définis à partir de 4 entiers :

- β l'entier ($\beta \geq 2$) définissant la base ;
- p le nombre de chiffres de la mantisse ;
- e_{min} l'exposant minimum et e_{max} l'exposant maximum.

correspondant à tous les nombres réels x s'écrivant :

$$x = s \left(\sum_{i=0}^{p-1} c_i \beta^{-i} \right) \beta^e, \text{ où } \begin{cases} s = \pm 1 \text{ le signe} \\ 0 \leq c_i \leq \beta - 1, \forall i \\ e_{min} \leq e \leq e_{max} \end{cases}$$

Exemple :

$$x = (1, 283 \cdot 10^3)_{10}$$

$s = +1$

$$x = (1 \times 10^0 + 2 \times 10^{-1} + 8 \times 10^{-2} + 3 \times 10^{-3}) \times 10^3$$

la représentation étant normalisée si $c_0 \neq 0$. **Rmq** : la partie mantisse pourra s'écrire $(c_0, c_1 \dots c_{p-1})_\beta$.

Définition des ensembles de flottants II

Pour représenter 0 on utilise une écriture spéciale en ne mettant que des 0 dans la mantisse (ce qui est logique) et un exposant de $e_{\min} - 1$.

$B=2$, $(c_0, c_1 c_2)$, e_{\min} , e_{\max}

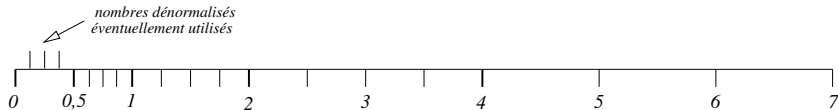
exercice : l'ensemble de flottants "jouets" $\mathbb{F}(2, 3, -1, 2)$

- 1 Trouver tous les nombres normalisés positifs (plus zéro) de $\mathbb{F}(2, 3, -1, 2)$
- 2 Les dessiner comme les traits d'une règle graduée.
- 3 Rajouter les (3) nombres dénormalisés positifs **non redondants**.
Exemple d'un nombre dénormalisé redondant : $(0, 10)_2 \times 2^1$ est une écriture (non normalisée) de 1 (avec notre convention l'écriture normalisée de 1 est $(1, 00) \times 2^0$).

Définition des ensembles de flottants III

Solution :

0		
$1,00 \ 2^{-1}$	=	$(0,5)_{10}$
$1,01 \ 2^{-1}$	=	$(0,625)_{10}$
$1,10 \ 2^{-1}$	=	$(0,750)_{10}$
$1,11 \ 2^{-1}$	=	$(0,875)_{10}$
$1,00 \ 2^0$	=	1
$1,01 \ 2^0$	=	$(1,25)_{10}$
\vdots	\vdots	\vdots
$1,11 \ 2^2$	=	$(7)_{10}$



Nombres dénormalisés non redondants : $(0,11)_2 \times 2^{-1} = (0,375)_{10}$,
 $(0,10)_2 \times 2^{-1} = (0,25)_{10}$, $(0,01)_2 \times 2^{-1} = (0,125)_{10}$,

Définition des ensembles de flottants III

Solution :

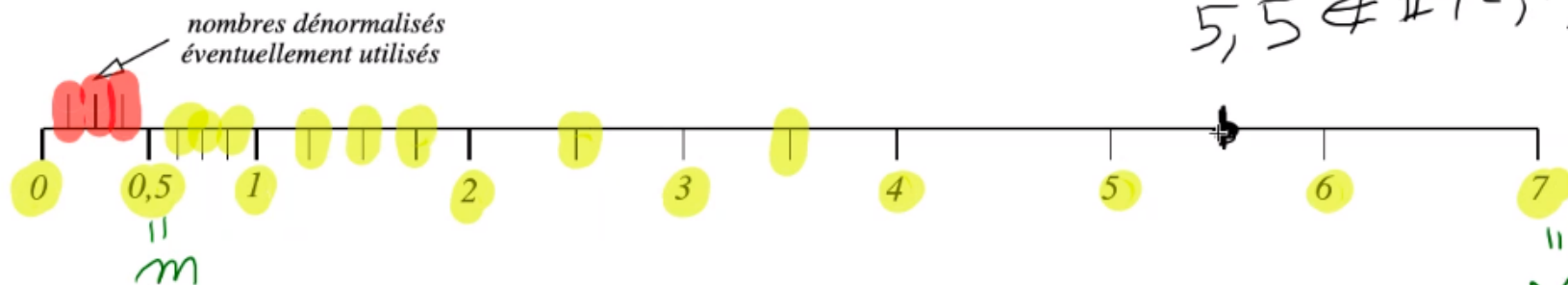
0		
1,00 2^{-1}	=	(0,5) ₁₀
1,01 2^{-1}	=	(0,625) ₁₀
1,10 2^{-1}	=	(0,750) ₁₀
1,11 2^{-1}	=	(0,875) ₁₀
1,00 2^0	=	1
1,01 2^0	=	(1,25) ₁₀
⋮	⋮	⋮
1,11 2^2	=	(7) ₁₀

$$= m = [1,00] \cdot 2^{-1}$$

" 0,5

$$= M$$

$$5,5 \notin \mathbb{F}/2, 3, -1, +2$$



Nombres dénormalisés non redondants : $(0,11)_2 \times 2^{-1} = (0,375)_{10}$,

$(0,10)_2 \times 2^{-1} = (0,25)_{10}$, $(0,01)_2 \times 2^{-1} = (0,125)_{10}$

Définition des ensembles de flottants IV

Remarques et notations :

- Dans les intervalles $[\beta^e, \beta^{e+1}]$ et $[-\beta^{e+1}, -\beta^e]$ l'incrément entre deux nombres flottants est constant et égal à β^{1-p+e} : pour obtenir le nombre suivant ou précédent, on ajoute/retranche

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

- Pendant assez longtemps on a utilisé uniquement les nombres normalisés (plus zéro) de ces ensembles. Cependant, comme la figure précédente le montre, il en résulte un “vide” entre le plus petit nombre normalisé et le zéro. L'utilisation des nombres dénormalisés non redondants (cf figure précédente) permet d'aller vers zéro plus graduellement.

Définition des ensembles de flottants V

- On notera M le plus grand nombre positif de $\mathbb{F}(\beta, p, e_{min}, e_{max})$:

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

m le plus petit nombre normalisé positif :

$$m = (1, 00..0) \beta^{e_{min}} = \beta^{e_{min}},$$

et μ le plus petit nombre dénormalisé (> 0) :

$$\mu = (0, 00..1) \beta^{e_{min}} = \beta^{1-p+e_{min}}.$$

- On notera \oplus , \ominus , \otimes , \oslash les opérations d'addition, de soustraction, de multiplication et de division, effectuées par l'ordinateur.

$$\begin{aligned}
 x &= (c_0, c_1 c_2 \dots c_{p-1})_\beta \beta^e \\
 0 &\leq c_i \leq \beta - 1 \\
 e_{\min} &\leq e \leq e_{\max} \\
 M &= ((\beta - 1), (\beta - 1)(\beta - 1) \dots (\beta - 1))_\beta \beta^{e_{\max}} \\
 &= (\beta - 1) (1, 1 \dots 1)_\beta \beta^{e_{\max}} \\
 (1, 1 \dots 1)_\beta &= 1 \times \beta^0 + 1 \times \beta^{-1} + \dots + 1 \times \beta^{-(p-1)} \\
 &= \sum_{i=0}^{p-1} \beta^{-i} \\
 M &= (\beta - 1) \left(\sum_{i=0}^{p-1} \beta^{-i} \right) \beta^{e_{\max}}
 \end{aligned}$$

Définition des ensembles de flottants V

- On notera M le plus grand nombre positif de $\mathbb{F}(\beta, p, e_{\min}, e_{\max})$:

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

m le plus petit nombre normalisé positif :

$$m = (1, 00 \dots 0)_\beta \beta^{e_{\min}} = \beta^{e_{\min}},$$

et μ le plus petit nombre dénormalisé (> 0) :

$$\mu = (0, 00 \dots 1)_\beta \beta^{e_{\min}} = \beta^{1-p+e_{\min}}.$$

$\nwarrow 1 \times \beta^{-(p-1)} = \beta^{1-p}$

- On notera $\oplus, \ominus, \otimes, \oslash$ les opérations d'addition, de soustraction, de multiplication et de division, effectuées par l'ordinateur.

$$\sum_{i=0}^{p-1} \beta^{-i} = ?$$

nb de
termes de la somme
 $n+1$

Somme géométrique
de raison q

$$\sum_{i=0}^n q^i = \frac{1-q}{1-q}$$

$$q = \beta^{-1} = \frac{1}{\beta} ; \quad \beta^{-i} = (\beta^{-1})^i$$

$$\begin{aligned} \sum_{i=0}^{p-1} \beta^{-i} &= \frac{1 - (\beta^{-1})^p}{1 - \beta^{-1}} = \frac{1 - \beta^{-p}}{1 - \frac{1}{\beta}} \\ &= \frac{1 - \beta^{-p}}{\left(\frac{\beta - 1}{\beta}\right)} = \frac{\beta}{\beta - 1} (1 - \beta^{-p}) \end{aligned}$$

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

$$M = \beta (1 - \beta^{-p}) \beta^{e_{\max}}$$

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

Définition des ensembles de flottants VI

- Dans les systèmes flottants actuels on rajoute des nombres spéciaux comme $+inf$, $-inf$ (inf comme infini) et NaN (pour Not a Number) qui ont une représentation spéciale (utilisant en particulier un exposant de $e_{max} + 1$). Enfin, on notera que, du fait du bit de signe, le zéro a deux représentations¹ qui conduisent à des résultats différents sur quelques calculs (par exemple $1 \oslash +0$ donnera $+inf$ alors que $1 \oslash -0$ donnera $-inf$).

1. que l'on notera $+0$ et -0 : si mathématiquement c'est le même nombre, informatiquement non !

Approximation d'un réel par un flottant I

Étant donné $x \in \mathbb{R}$, en général $x \notin \mathbb{F}(\beta, p, e_{min}, e_{max})$ et on associe à x une approximation $fl(x) \in \mathbb{F}(\beta, p, e_{min}, e_{max})$ avec le critère suivant :

$fl(x)$ est le flottant le plus proche de x et si x est à égale distance de deux flottants celui dont le dernier chiffre est pair est choisi (à partir de maintenant on suppose que β est pair).

Attention cependant cette règle ne fonctionne pas pour les nombres de magnitude supérieure à M : en toute rigueur il faut considérer d'abord $\tilde{fl}(x)$ comme étant la même opération mais à valeur dans $\mathbb{F}(\beta, p, e_{min}, +\infty)$ et si :

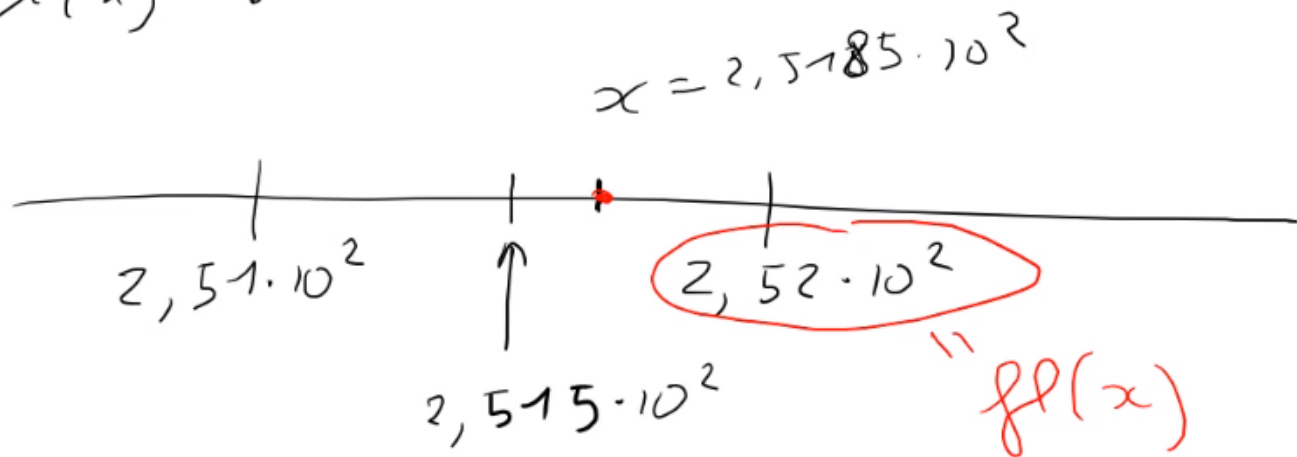
- $|\tilde{fl}(x)| > M$ alors $fl(x)$ est le nombre flottant spécial $sign(x)inf$: on dit qu'il y a **“overflow”** !
- et sinon $fl(x) = \tilde{fl}(x)$.

Example

$$\mathbb{F}(10, 3, -3, +3)$$

$$* x = \underline{2,5185} \cdot 10^2 \notin \mathbb{F}(10, 3, -3, +3)$$

$$fl(x) ?$$



$$* x = 2,515 \cdot 10^2 \notin \mathbb{F}(10, 3, -3, +3)$$

$$2,5\underline{2} \cdot 10^2$$

↖ chiffre pair

$$\Rightarrow fl(x) = 2,52 \cdot 10^2$$

$$* fl(2,525 \cdot 10^2) = 2,52 \cdot 10^2$$

Autre exemple dans le cas $|x| > M$; $\beta = 2$

\sim
 \tilde{M}



$$M = \underbrace{(1, 1 \dots 1)}_p \cdot 2^{e_{\max}} \text{ dans } \mathbb{F}(2, p, e_{\min}, e_{\max})$$

On considère le système $\mathbb{F}(2, p, e_{\min}, +\infty)$

et $M' \in \mathbb{F}(2, p, e_{\min}, +\infty)$ qui vient juste après M

$$M' = (1, 0 \dots 0) 2^{e_{\max} + 1} \quad (\text{écriture normalisée})$$

$$\tilde{M} = \frac{M + M'}{2}$$



Si $|x| > \tilde{M}$ alors $fl(x) = +\infty$

Si $M \leq |x| < \tilde{M}$ alors $fl(x) = M$

Approximation d'un réel par un flottant II

Seuil d'overflow : avec cette règle M ne correspond pas exactement au seuil d'overflow. Dans $\mathbb{F}(\beta, p, e_{min}, +\infty)$ le nombre flottant juste après M est $\beta^{e_{max}+1}$, dont le dernier chiffre de la mantisse est pair. Ainsi le seuil d'overflow est exactement :

$$\tilde{M} := \frac{M + \beta^{e_{max}+1}}{2} = M + \frac{\beta}{2} \beta^{-p} \beta^{e_{max}}$$

mais du fait de la règle d'arrondi $\tilde{fl}(\tilde{M}) = \beta^{e_{max}+1}$ (et pas M) et donc $fl(\tilde{M}) = Inf$. Ainsi tout nombre réel x tel que $|x| < \tilde{M}$ doit être codé par un flottant “usuel” (cad qui n'est pas un nombre flottant spécial comme $\pm Inf$ ou Nan).

Approximation d'un réel par un flottant III

Epsilon machine : On montre que si $|x| \in [m, \tilde{M}[$ alors l'erreur relative est bornée par un nombre appelé epsilon machine :

$$\frac{|fl(x) - x|}{|x|} \leq \mathbf{u} := \frac{\beta^{1-p}}{2}.$$

Preuve : Soit donc $x \in \mathbb{R}$ tel que $|x| \in [m, \tilde{M}[$. Il existe $e \in \llbracket e_{min}, e_{max} \rrbracket$ tel que $|x| \in [\beta^e, \beta^{e+1}[$. Nous avons vu que l'incrément entre deux flottants dans la plage $[\beta^e, \beta^{e+1}[$ est égal à β^{1-p+e} , ainsi l'erreur absolue est bornée par la moitié de cette quantité (approximation par le flottant le plus proche) :

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

Approximation d'un réel par un flottant IV

On peut alors borner l'erreur relative en divisant par le plus petit nombre de la plage, d'où :

$$er = \frac{|x - fl(x)|}{|x|} \leq \frac{ea}{\beta^e} = \frac{1}{2}\beta^{1-p} \quad \square$$

Une façon plus pratique de noter cette erreur relative est d'écrire que (cf TD) :

$$fl(x) = x(1 + \epsilon), \text{ avec } |\epsilon| \leq \mathbf{u}$$

Lorsque $|x| \in [0, m[$ avec $fl(x) \neq x$ on dit qu'il y a **“underflow”**. Dans ce cas l'erreur relative n'est plus maîtrisée. On passe d'une erreur relative quasi bornée par \mathbf{u} au voisinage de $|m|$ à une erreur relative de 1 (pour les nombres non nuls très proches de 0 et qui seront codés par 0 (c'est à dire tous les nombres réels dans $[-\mu/2, \mu/2]$), excepté 0).

Règles des 4 opérations usuelles I

Les 4 opérations usuelles sur les flottants doivent respecter le critère suivant : tout se passe comme si le calcul était exact puis arrondi au flottant le plus proche, c'est à dire que si \cdot est l'une des 4 opérations et \odot l'opération machine correspondante, et x et y sont deux nombres flottants alors :

$$x \odot y = fl(x \cdot y)$$

et donc en l'absence d'overflow et d'underflow on a :

$$x \odot y = (x \cdot y)(1 + \epsilon), \text{ avec } |\epsilon| \leq u$$

Remarques :

- La norme sur les flottants impose aussi cette précision pour la racine carrée.

Règles des 4 opérations usuelles II

- On peut montrer que la soustraction entre deux flottants x et y de même signe est exacte s'ils sont de magnitude voisine, plus exactement :

Soient deux flottants x et y de même signe et tels que $x \leq y \leq 2x$ ou $y \leq x \leq 2y$ alors :

$$x \ominus y = x - y$$

Rmq : Si la soustraction de deux flottants voisins est exacte, elle est quand même à l'origine de certains problèmes de perte de précision car les nombres x et y résultent le plus souvent de calculs qui comporteront des erreurs et la soustraction amplifiera ces mêmes erreurs (cf TD).

Flottants utilisés sur ordinateurs I (norme IEEE754)

Les deux ensembles de flottants les plus utilisés sur ordinateurs sont :

- 1 $\mathbb{F}(2, 24, -126, 127)$ appelés flottants “simple précision” dont le codage tient sur 4 octets,
- 2 $\mathbb{F}(2, 53, -1022, 1023)$ appelés flottants “double précision” tenant sur 8 octets.

et voici leurs nombres caractéristiques (valeurs approchées) :

ensembles de flottants	$\mathbb{F}(2, 24, -126, 127)$	$\mathbb{F}(2, 53, -1022, 1023)$
$u \simeq$	$5.96 \cdot 10^{-8}$	$1.11 \cdot 10^{-16}$
$m \simeq$	$1.17 \cdot 10^{-38}$	$2.225 \cdot 10^{-308}$
$M \simeq$	$3.40 \cdot 10^{38}$	$1.79 \cdot 10^{308}$

Remarque : sauf problème d’overflow ou d’underflow toute multiplication/division par une puissance de 2 est exacte avec ces nombres flottants.

Simple précision :

1 octet = 8 bits
4 octets = 32 bits

1 bit 26 bits 7 bits
designe mantisse exposant

Flottants utilisés sur ordinateurs I (norme IEEE754)

Les deux ensembles de flottants les plus utilisés sur ordinateurs sont :

- 1 $\mathbb{F}(2, 24, -126, 127)$ appelés flottants "simple précision" dont le codage tient sur 4 octets,
- 2 $\mathbb{F}(2, 53, -1022, 1023)$ appelés flottants "double précision" tenant sur 8 octets.

et voici leurs nombres caractéristiques (valeurs approchées) :

ensembles de flottants	$\mathbb{F}(2, 24, -126, 127)$	$\mathbb{F}(2, 53, -1022, 1023)$
$u \simeq$	$5.96 \cdot 10^{-8}$	$1.11 \cdot 10^{-16}$
$m \simeq$	$1.17 \cdot 10^{-38}$	$2.225 \cdot 10^{-308}$
$M \simeq$	$3.40 \cdot 10^{38}$	$1.79 \cdot 10^{308}$

Remarque : sauf problème d'overflow ou d'underflow toute multiplication/division par une puissance de 2 est exacte avec ces nombres flottants.

$M > 2^{1023}$

$M = (1, 1000 \dots 1)_2 \cdot 2^{1023} > 2^{1023}$

53

Un exemple de calcul d'erreur I

Pour faire une analyse de précision de calculs menés avec les flottants, on fait l'hypothèse *qu'il n'y a pas eu d'overflow ni d'underflow*.

Exemple : on considère a , b et c des flottants et on veut analyser la précision de $x_c := (a \otimes b) \oslash (c \otimes d)$. Si on fait cette hypothèse alors :

$$a \otimes b = (a \times b)(1 + \epsilon_1), \quad |\epsilon_1| \leq u$$

$$c \otimes d = (c \times d)(1 + \epsilon_2), \quad |\epsilon_2| \leq u$$

$$(a \otimes b) \oslash (c \otimes d) = \frac{(a \otimes b)}{(c \otimes d)}(1 + \epsilon_3), \quad |\epsilon_3| \leq u$$

soit finalement :

$$x_c = \underbrace{\frac{a \times b}{c \times d}}_x \underbrace{\frac{(1 + \epsilon_1)(1 + \epsilon_3)}{(1 + \epsilon_2)}}_{(1+\delta)}$$

Un exemple de calcul d'erreur II

où, comme $\mathbf{u} \ll 1$, l'erreur relative $|\delta|$ est “quasi-bornée” par $3\mathbf{u}$. Une manière élégante de traiter ce problème est d'utiliser le résultat suivant :

lemme de simplification

Si $|\epsilon_k| \leq \mathbf{u}$, $s_k = \pm 1$ pour $k = 1, \dots, n$ et si $n\mathbf{u} < 1$ alors :

$$\prod_{k=1}^n (1 + \epsilon_k)^{s_k} = 1 + \delta_n \quad \text{avec} \quad |\delta_n| \leq \frac{n\mathbf{u}}{1 - n\mathbf{u}}$$

Preuve : cf TD 1.

En utilisant ce lemme dans l'exemple précédent, on obtient donc que :

$$|\delta| \leq \frac{3\mathbf{u}}{1 - 3\mathbf{u}} \simeq 3\mathbf{u}$$

Un exemple de calcul d'erreur III

Rmq : En général les bornes ainsi obtenues sont (bien) plus grandes que l'erreur relative “exacte” (il faut pour s'en approcher que chaque opération soit réalisée avec l'erreur maximum et qu'il n'y ait aucune compensation dans les termes en ϵ_k).