Université de Picardie Jules Verne
Licence 2
Scilab
2017-2018

# sur l'arithmétique de l'ordinateur

## 1 Représentation des réels en virgule flottante

### 1.1 La représentation

On rappelle tout d'abord le résultat bien connu : Soit $b \in \mathbb{N} - \{0, 1\}$. Soit $x \in \mathbb{R}^*$. Alors $x$ se représente de manière unique par

$$x = \pm M.\ b^E$$

où M est la mantisse normalisée ($\frac{1}{b} \leq M < 1$) et où E est l'exposant (entier).

On peut écrire

$$M = \sum_{i=1}^{+\infty} a_i b^{-i}\ , 0 \leq a_i < b,\ a_1 \neq 0, \tag{1}$$

ainsi

$$x = \pm 0, a_1 a_2 a_3 \cdots a_n \cdots \quad b^E$$

Bien évidemment, les nombres réels représentables sur ordinateur ne peuvent s'obtenir qu'en tronquant le développement (**??**). Les réels représentés seront donc de la forme :

$$\bar{x} = \pm 0, a_1 a_2 \cdots a_r \quad b^E$$

où

$r$ est le nombre de chiffres significatifs
E est l'exposant,
$a_1 \cdots a_r$ est la mantisse

On peut dès lors définir l'ensemble $F(b, r, m_1, m_2)$ des nombres en virgule flottante par

$$F(b, r, m_1, m_2) = \{\pm 0, a_1 \cdots a_r b^j\ ;\ m_1 \leq j \leq m_2\}$$

La normalisation (mantisse $1 \leq M < b$) est fondamentale : tous les nombres en virgule flottante auront ainsi le même nombre de chiffres significatifs. Ils seront donc codés sur un même type de support.

# 2 Erreurs en précision finie

## 2.1 Erreurs de représentation

### 2.1.1 Quelques définitions

En précision finie, on remplace le réel $x$,

$$x = \pm \left( \sum_{i=1}^{+\infty} a_i b^{-i} \right) b^E$$

par

$$\bar{x} = \pm \left( \sum_{i=1}^{m} x_i b^{-i} \right) b^E.$$

$|x - \bar{x}|$ est l'erreur absolue due à la représentation

$\frac{|x-\bar{x}|}{|x|}$ est l'erreur relative ——————— ———

La mantisse à $m$ chiffres peut se déduire de la mantisse exacte de quatre manières parmi lesquelles :

- par troncature
$$x_i = a_i \ ; \ i = 1, \cdots, m.$$
- par arrondi au plus proche
$$x_i = a_i \ ; \ i = 1, \cdots, m-1, \ x_m = \begin{cases} a_m & \text{si } a_{m+1} \in [0, \frac{b}{2}] \\ a_m + 1 & \text{si } a_{m+1} \in [\frac{b}{2}, b[ \end{cases}$$

La fonction d'arrondi $A$ devra avoir les propriétés suivantes

- $A$ est définie sur tout $\mathbb{R}$
- $A$ laisse invariant $F(b, r, m_1, m_2)$
- Soit $x \in \mathbb{R}$ et $[f, f']$ le plus petit intervalle contenant $x$ tel que $f, f' \in F(b, r, m_1, m_2)$,
$$\text{Alors } A(x) = \begin{cases} f & \text{si } |x - f| < |x - f'| \\ f' & \text{sinon} \end{cases}$$
- si $x = \frac{f + f'}{2}$ alors $A(x)$ sera déterminé de diverses façons, généralement liées à la machine.

### 2.1.2 Quelques propriétés

Soit $\mathcal{F} = F(b, r, m_1, m_2)$ un ensemble de réel en virgule flottante. On définit l' "$\epsilon$ machine" (ou ULP : Unit in the Last Place) $\tau$ comme le plus petit réel positif de $\mathcal{F}$ tel que $1 + \tau$ est le plus petit réel supérieur à 1 et qui sera représenté par un nombre de $\mathcal{F}$ différent de 1. Lorsque l'exposant est limité à un intervalle $[-N_1, N_2]$, $N_i > 0$, le plus petit réel représentable est

$$r = b^{-1-N_2} (\text{ "seuil d'underflow" })$$

et le plus grand entier représentable est

$$R = (1 - b^{-r})b^{N_2} \sim b^{N_2}(\text{ "seuil d'overflow" }).$$

Ici r est le nombre de chiffres significatifs. L'erreur relative due à la représentation est majorée par

$b^{1-r}$ (pour la troncature) et
$\frac{1}{2}b^{1-r}$ (pour l'arrondi)

### 2.1.3 Erreurs dans les opérations arithmétiques

Soient $x$ et $y$ deux réels, $\bar{x}$ et $\bar{y}$ leur représentation dans le système $\mathcal{F} = F(b, r, m_1, m_2)$. On pose

$$\delta x = x - \bar{x}, \quad \delta y = y - \bar{y}$$

Soit alors $\bullet$ une opération sur les réels et $\circ$ l'opération correspondante sur les flottants. On a en particulier

$$\delta(x \bullet y) = x \bullet y - \bar{x} \circ \bar{y}$$

On a le résultat suivant : Avec les notations précédentes, on a

$$\begin{array}{ll}
\delta(x+y) = \delta(x) + \delta(y) & + \text{ terme négligeable} \\
\delta(x-y) = \delta(x) - \delta(y) & + \text{ terme négligeable} \\
\dfrac{\delta(x \times y)}{(x \times y)} = \dfrac{\delta(x)}{x} + \dfrac{\delta(y)}{y} & + \text{ terme négligeable} \\
\dfrac{\delta(x/y)}{(x/y)} = \dfrac{\delta(x)}{x} + \dfrac{\delta(y)}{y} & + \text{ terme négligeable}
\end{array}$$

Exemple
Voici ce qui peut arriver dans $F(10, 3, ., .)$ lorsqu'on effectue l'opération `1.02 - 0.0617`

| nb de chiffres de garde | 0 | 1 |
|---|---|---|
| Dénormalisation | 0.102 $10^1$<br>0.00617 $10^1$ | 0.1020 $10^1$<br>0.0061 7 $10^1$ |
| Résultat intermédiaire | 0.096 $10^1$ | 0.0959 $10^1$ |
| Normalisation | 0.96 $10^0$ | 0.959 $10^0$ |

## 2.2 Perte de précision et propagation d'erreur d'arrondi

Chaque opération en virgule flottante génère une erreur qui peut être amplifiée dans les opérations suivantes.

### 2.2.1 Erreur de cancellation

L'erreur de cancellation intervient lorsque l'on soustrait deux nombres $x$ et $y$ proches.

L'approximation de $z = x - y$ sera donnée par le nombre en virgule flottante : $\bar{z} = \bar{x} - \bar{y}$. Ainsi, $\bar{z}$ aura $r$ chiffres significatifs exacts si et seulement si $\bar{x}$ et $\bar{y}$ n'ont pas de chiffres en commun.

Exemple 1 Soient

$$\bar{x} = 0,76545421 \ \ 10^1 \text{ et}$$
$$\bar{y} = 0,76544200 \ \ 10^1.$$

Si l'on approche $x$ et $y$ avec 7 chiffres corrects, on aura

$$\bar{z} = \bar{x} - \bar{y} = 0,12210000 \ \ 10^{-3} \ .$$

L'approximation de $z$, $\bar{z}$ n'a que 3 chiffres corrects puisque le quatrième s'obtient des huitièmes chiffres de $\bar{x}$ et $\bar{y}$.
L'erreur relative est 10000 fois plus grande que l'erreur relative de $\bar{x}$ et de $\bar{y}$.

$\Longrightarrow$ *éviter de soustraire des quantités voisines.*

Exemple 2 On veut calculer $f(x) = 1 - \cos(x)$ au voisinage de 0 avec une arithmétique de 6 chiffres décimaux.

Comme $\cos(x) \simeq 1$, pour $x \simeq 0$ on aura une perte de chiffres significatifs (l'erreur sera ici de l'ordre de grandeur de $f$ au voisinage de 0).

Il faut donc utiliser une autre formule : $f(x) = \dfrac{\sin^2(x)}{1 + \cos(x)}$.

Exercice : Programmer ces deux formules (en simple précision) pour x=`1.E-2, 1.E-3` .

### 2.2.2 Effet de cumul

Il apparaît lors de l'addition de nombres ayant des ordres de grandeur différents.

Exemple Plaçons-nous dans une arithmétique de 4 chiffres. Soient $x = 1.145$ et $y = 0,3112 \ \ 10^{-1}$.

| | | | |
|---|---|---|---|
| x | 0.1145 | 00 | $10^1$ |
| y | 0.0031 | 12 | $10^1$ |
| x+y | 0.1176 | 00 | $10^1$ |

On observe donc une perte de chiffres significatifs. C'est typiquement ce qui arrive lorsque l'on fait un cumul simple i.e.

$$S_{n+1} = S_n + a_{n+1}$$

A partir d'un certain rang, n, $a_{n+1}$ deviendra petit devant $S_n$ et à partir donc d'un certain rang (procédure de dénormalisation), on aura

$$S_{n+1} = (S_n + a_{n+1}) = S_n$$

$\Longrightarrow$ *regrouper les calculs par ordres de grandeur semblables*

# 3 Quelques critères pratiques

## 3.1 Calcul de série

### 3.1.1 Critères analytiques

Il faut évidemment, avant tout, exploiter les propriétés éventuelles de la série $\sum_{i=1}^{+\infty} a_i$.

Si par exemple la série est alternée ($a_i = (-1)^i b_i,\ b_i > 0,\ \ b_{i+1} < b_i, \forall i \geq 1$), on vérifie aisément que $|S - S_n| \leq b_{n+1}$. On pourra alors déterminer à l'avance, pour $\epsilon$ donné, la valeur de n telle que $|S - S_n| < \epsilon$.

### 3.1.2 Critères numériques

Bien évidemment, on ne dispose pas toujours de critère d'arrêt et *a fortiori* pas d'estimation d'erreur (ce qui implique que l'on somme à l'endroit en général). On peut néanmoins arrêter les itérations lorsque la quantité ajoutée est considérée comme "petite" i.e. si

$$|a_n| < \epsilon \text{ ou } \frac{|a_n|}{|S_n|} < \epsilon$$

Si on dispose d'une estimation d'erreur (de $|S - S_n|$), il convient de sommer la série "à l'envers", pour éviter les effets de cumul, comme le montre l'exemple suivant :
Application : $S_n = 1 - \frac{1}{2} + \frac{1}{3} + \cdots + \frac{(-1)^{n+1}}{n}$. Calculer (en simple précision) cette série en partant du début, en partant de la fin, avec une précision de $10^{-6}$ et comparer le résultat obtenu avec `log(2.0)`.

## 3.2 Les suites

### 3.2.1 Critères analytiques

Même remarque que pour le cas des séries.

### 3.2.2 Critères numériques

On peut considérer les critères $|x_{n+1} - x_n| < \epsilon$ ou $\frac{|x_{n+1} - x_n|}{|x_n|} < \epsilon$

### 3.2.3 Autres tests

Lorsque l'on met en œuvre une méthode itérative, il faut se donner un nombre maximal d'itérations au dela duquel on décide qu'il est inutile de poursuivre le processus. En effet, dans le cas contraire, une erreur de programmation peut conduire à une boucle effectuée indéfiniment.

## 3.3 Contrôles *a posteriori*

Prendre des précautions dans la construction du programme ne suffit pas toujours. Il faut pouvoir analyser les résultats obtenus, ce qui permet de detecter des erreurs (s'il y en a).

### 3.3.1 Nombre de chiffres exacts

Soit $x_n$ une suite convergente vers $x^*$. On mesure le nombre de chiffres décimaux exacts par

$$e(x_n, x^*) = \log_{10}\left(\frac{|x^*|}{|x_n - x^*|}\right).$$

Le nombre de chiffres exacts gagnés de $x_n$ à $x_{n+1}$ est alors donné par

$$d_n = e(x_{n+1}, x^*) - e(x_n, x^*) = \log_{10}\frac{|x_n - x^*|}{|x_{n+1} - x^*|}.$$

Si $x_n$ est une suite d'ordre $r$ i.e. si

$$0 < \underline{lim}\frac{|x_{n+1} - x^*|}{|x_n - x^*|^r} \leq \overline{lim}\frac{|x_{n+1} - x^*|}{|x_n - x^*|^r} < +\infty$$

alors le nombre de chiffres exacts est à peu près multiplié par $r$ à chaque itération.
Exercice : comparer les résultats intermédiaires dans les exercices 1 et 2, page 18 du poly.

### 3.3.2 Quantités proches de l'$\epsilon$ machine

Il convient d'être prudent quant à la validité de résultats numériques dont l'ordre de grandeur est de l'ordre ou inférieur à l'$\epsilon$ machine : la précision affichée est souvent illusoire.

### 3.3.3 Mélange des types

Attention : mélanger les types de variables peut conduire à des résultats numériques absurdes.
Exemple Dans l'exercice 1, page 18, déclarer la fonction $f$ en simple précision.

On pourra consulter avec profit les ouvrages suivants :

**J.-M. Muller**
*Arithmétique des ordinateurs*
*opérateurs et fonctions élémentaires*
Coll. Etudes et recherches en informatique
Masson, Paris, 1989

**J.-C. Bajard, O. Beaumont, J.-M. Chesneaux, M. Daumas, J. Erhel, D. Michelucci, J.-M. Muller, B. Philippe, N. Revol, J.-L. Roch, J. Vignes**
coordonné par M. Daumas et J.-M. Muller
*Qualité des calculs sur ordinateurs*
Masson, 1997.