Introduction au calcul scientifique

christophe.labourdette(at)cmla.ens-cachan.fr

Septembre 2016

Le bit

description

- Une unité fondamentale : le bit
- Un booléen, deux états : le 0 (faux) et le 1 (vrai)
- les opérations booléennes, binaires (*ou*,$\vee$), (*et*,$\wedge$), négation (*not*,$\neg$)

opération

x	y	$x \vee y$	$x \wedge y$	$\neg x$
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

représentation simple

Si on considère une suite de bits $d_{N-1}d_{N-2}\cdots d_2 d_1 d_0$ de longueur N, comme la représentation en base 2 d'un entier positif, c'est-à-dire :

$$d_{N-1}d_{N-2}\cdots d_2 d_1 d_0 \doteq \sum_{j=0}^{N-1} d_j 2^j \ , \ d_j \in \{0,1\}$$

On couvre alors l'ensemble des entiers naturels de 0 à $2^N - 1$:

$$\begin{aligned} 0000\cdots0000 &\doteq 0 \\ 0000\cdots0001 &\doteq 1 \\ &\vdots \\ 1111\cdots1111 &\doteq 2^N - 1 \end{aligned}$$

$\mathbb{Z}$

Pour étendre la représentation à $\mathbb{Z}$, on utilise un bit $s \in \{0, 1\}$ (appelé bit de signe) pour représenter le signe de l'entier.
Notre suite de N bits devient alors :

$$sd_{N-2} \cdots d_2 d_1 d_0 \doteq (-1)^s \sum_{j=0}^{N-2} d_j 2^j$$

Le domaine couvert est donc $[-(2^{N-1} - 1), 2^{N-1} - 1] \subset \mathbb{Z}$
Mais il y a des problèmes.

problème d'unicité et de continuité

$$
\begin{aligned}
0000\cdots0000 &\doteq 0\\
0000\cdots0001 &\doteq 1\\
&\vdots\\
0111\cdots1111 &\doteq 2^{N-1}-1\\
1000\cdots0000 &\doteq -0\\
1000\cdots0001 &\doteq -1\\
&\vdots\\
1111\cdots1111 &\doteq -(2^{N-1}-1)
\end{aligned}
$$

Codage en complément à un

En utilisant simplement un bit de signe, nous obtenons donc un -0 et un $+0$.
Pour essayer de régler ce problème, on introduit donc le codage en complément à un pour représenter les entiers négatifs comme suit :

$$\bar{x} = \sum_{j=0}^{N-1} (1 - d_j) 2^j$$

pour représenter $-x$.

Codage en complément à un (2)

$$
\begin{array}{lcl}
0000\cdots0000 & \doteq & 0 \\
0000\cdots0001 & \doteq & 1 \\
 & \vdots & \\
0111\cdots1111 & \doteq & 2^{N-1}-1 \\
1000\cdots0000 & \doteq & -(2^{N-1}-1) \\
 & \vdots & \\
1111\cdots1110 & \doteq & -1 \\
1111\cdots1111 & \doteq & -0
\end{array}
$$

Codage en complément à deux

En utilisant le complément à un, le codage de zéro n'est toujours pas unique mais l'ordre des entiers est plus satisfaisant. Il est encore plus naturel si l'on considère le codage en complément à deux avec :

$$\bar{\bar{x}} = 1 + \sum_{j=0}^{N-1} (1 - d_j) 2^j$$

pour représenter $-x$:

Codage en complément à deux (2)

$$
\begin{array}{lcl}
0000\cdots0000 & \doteq & 0 \\
0000\cdots0001 & \doteq & 1 \\
 & \vdots & \\
0111\cdots1111 & \doteq & 2^{N-1}-1 \\
1000\cdots0000 & \doteq & -2^{N-1} \\
 & \vdots & \\
1111\cdots1110 & \doteq & -1 \\
1111\cdots1111 & \doteq & -0
\end{array}
$$

C'est cette dernière représentation qui est utilisée sur les processeurs Intel. Le codage du zéro devient unique mais l'intervalle des entiers représentés $[-2^{N-1}, 2^{N-1}-1]$ n'est plus symétrique.

Nombres flottants

Pour le calcul scientifique les entiers relatifs ne sont pas suffisant, il faut bien entendu un type pour représenter les réels. Le problème est beaucoup plus compliqué que pour les entiers. On va considérer que chaque suite de bits de longueur fixé représentant un flottant est séparée en trois champs :

- le signe $s \in \{0, 1\}$
- l'exposant e est un entier
- la mantisse m est un réel positif du type

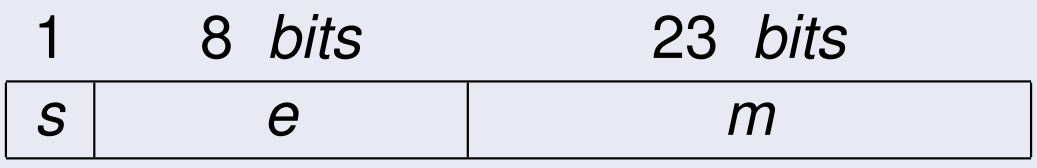
$$m = d_1 b^{-1} + d_2 b^{-2} + \cdots + d_p b^{-p}$$

Le nombre réel est donné par $x = (-1)^s b^e m$
Cette représentation à "virgule flottante" est ce que les fabricants de calculatrice appellent la "notation scientifique".

En général on trouve deux formats :

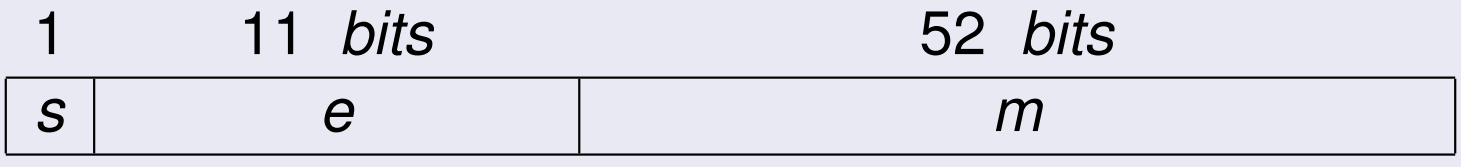
simple précision

La simple précision, $N = 32$ bits



double précision

La double précision, $N = 64$ bits



$\mathbb{F}$

L'ensemble des nombres réels qui peuvent être représentés par le codage flottant est noté $\mathbb{F}$ et dépend du choix de la base b, de la mantisse m et des valeurs possibles de e.
Un système de nombres flottants en base b, de précision p et comportant les exposants $[e_{min}, e_{max}] \subset \mathbb{Z}$ contient donc les réels suivants :

$$x = (-1)^s b^e \sum_{j=1}^{p} d_j b^{-j}$$

où

$$s \in \{0, 1\}$$
$$e \in \{e_{min}, e_{min} + 1, \cdots, e_{max}\}$$
$$d_j \in \{0, 1, \cdots, b-1\}, \quad j = 1, 2, \cdots, p$$

Les valeurs d_j représente les *digits* de la mantisse
$m = d_1 b^{-1} + d_2 b^{-2} + \cdots + d_p b^{-p}$.

Normalisation

Chaque valeur des paramêtres s,e et $d_1, d_2, \cdots, d_p$ représente un nombre réel.
Mais plusieurs jeux de paramêtres différents peuvent représenter le même nombre réel x.
Pour construire un système possédant une représentation de réels unique sans réduire l'amplitude du système on introduit la condition $d_1 \neq 0$.
Les nombres alors obtenus avec $b^{-1} \leq m < 1$ sont dit *normalisés*.
Dans la suite l'ensemble des nombres normalisés obtenus, étendus avec 0 (spécifié par la mantisse $m = 0$), sont notés $\mathbb{F}_N$.

Normalisation (2)

La restriction $d_1 \neq 0$ nous donne les valeurs :

$$
\begin{aligned}
& s \in \{0, 1\} \\
& e \in \{e_{min}, e_{min} + 1, \cdots, e_{max}\} \\
& d_1 \in \{1, \cdots, b-1\}, \\
& d_j \in \{0, 1, \cdots, b-1\}, \quad j = 2, \cdots, p
\end{aligned}
$$

Ainsi que les nombres dans $\mathbb{F}_N \backslash \{0\}$

Normalisation(3)

Les nombres perdus par la condition de normalisation $d_1 \neq 0$ peuvent être récupérés, sans perdre l'unicité de la représentation, en autorisant $d_1 = 0$ uniquement pour le plus petit exposant $e = e_{min}$.
Les nombres avec $m \in [b^{-p}, b^{-1} - b^{-p}]$ qui sont les les éléments de l'intervalle $(-b^{e_{min}-1}, b^{e_{min}-1})$ sont dit *dénormalisés* (ou *sous-normaux*) et leur ensemble est noté $\mathbb{F}_D$.
Le nombre zéro (qui correspond à la mantisse $m = 0$) est un élément de $\mathbb{F}_N$ d'après la définition précédente.
Pour garantir l'unicité de la représentation il faut faire un choix pour le signe, en général $s = 0$.

Un système de nombres flottants

Un système de nombres flottants est donc basé sur les paramêtres suivants :

- La base $b \geq 2$
- La précision $p \geq 2$
- Le plus petit exposant $e_{min} < 0$
- Le plus grand exposant $e_{max} > 0$
- L'indicateur de normalisation qui est un Booleen

La valeur booléenne *denorm* indique l'utilisation ou non d'un système de normalisation.

En pratique

Dans la suite on notera donc pour le système de représentation des nombres flottants : $\mathbb{F}(b, p, e_{min}, e_{max}, denorm)$
Il satisfait aux relations suivantes :

$$\begin{aligned} \mathbb{F}(b, p, e_{min}, e_{max}, vrai) &= \mathbb{F}_N(b, p, e_{min}, e_{max}) \cup \\ &\quad \mathbb{F}_D(b, p, e_{min}, e_{max}) \\ \mathbb{F}(b, p, e_{min}, e_{max}, faux) &= \mathbb{F}_N(b, p, e_{min}, e_{max}) \end{aligned}$$

Les systèmes habituellement utilisés par les microprocesseurs Intel sont respectivement $\mathbb{F}(2, 24, -125, 128, vrai)$ et $\mathbb{F}(2, 53, -1021, 1024, vrai)$ pour les représentations en simple et en double précision. Il existe aussi pour ces mêmes microprocesseurs un système de précision étendue $\mathbb{F}(2, 64, -16381, 16384, vrai)$

Exemples

L'exposant est repésenté avec un biais, ici par exemple on a E+127. Le nombre $1 = (1.000\cdots0)_2 \times 2^0$ est stocké comme suit :

0	01111111	00000000000000000000000

Le nombre $11/2 = (1.011)_2 \times 2^2$ est représenté par :

0	10000001	01100000000000000000000

Le nombre $1/10 = (1.100110011\cdots)_2 \times 2^{-4}$ a une expression binaire infinie si on la tronque à la taille de la mantisse on trouve :

0	01111011	10011001100110011001100

Un sous ensemble fini des réels

On considère un système de calcul flottant du type

$$\mathbb{F}(b, p, e_{min}, e_{max}, denorm)$$

Il est évident que l'ensemble $\mathbb{F}$ est un sous-ensemble fini des nombres réels.
Il est alors naturel de se poser quelques questions sur $\mathbb{F}$:

- De quelle taille est $\mathbb{F}$?
- Quelles sont les bords de $\mathbb{F}$?
- Quelle est la distance entre deux élements de $\mathbb{F}$?

Dimension de $\mathbb{F}$

La dimension de l'ensemble de nombres normalisés est

$$2(b-1)b^{p-1}(e_{max}-e_{min}+1)$$

Bien entendu on n'a pas tenu compte du nombre zéro et de l'ensemble des nombres normalisés si la variable *denorm* est vraie. Dans le cas des systèmes classiques on trouve donc : pour $\mathbb{F}(2,24,-125,128,\mathit{vrai})$ et $\mathbb{F}(2,53,-1021,1024,\mathit{vrai})$ respectivement

$$2^{24}.254 \approx 4.26.10^{9} \text{ et } 2^{53}.2046 \approx 1.84.10^{19}$$

nombres normalisés.

Extrémums de $\mathbb{F}$

Il est important de noter que contrairement à l'ensemble des nombres réels qu'il sont censés représenter, les ensembles $\mathbb{F}$, qui sont finis possèdent un maximum x_{max} et un plus petit nombre positif et normalisé x_{min}.
Avec les valeurs $m = m_{max}$ et $e = e_{max}$, le plus grand des nombres flottants est

$$x_{max} := \max\{x \in \mathbb{F}\} = (1 - b^{-p})b^{e_{max}}$$

Avec les valeurs $e = e_{min}$ et $m = m_{min}$, le plus petit nombre flottant positif et normalisé est

$$x_{min} := \min\{x \in \mathbb{F}_N : x > 0\} = b^{e_{min}-1}$$

Extrémums de $\mathbb{F}$ (2)

Dans les systèmes utilisés on trouve :
pour l'ensemble $\{x \in \mathbb{F}_N(2, 24, -125, 128) : x > 0)\}$

$$\begin{array}{rcccl} x_{min} & = & 2^{-126} & \approx & 1.18.10^{-38} \\ x_{max} & = & (1 - 2^{-24})2^{128} & \approx & 3.40.10^{38} \end{array}$$

et pour la double précision
$\{x \in \mathbb{F}_N(2, 53, -1021, 1024) : x > 0)\}$

$$\begin{array}{rcccl} x_{min} & = & 2^{-1022} & \approx & 2.23.10^{-308} \\ x_{max} & = & (1 - 2^{-53})2^{1024} & \approx & 1.80.10^{308} \end{array}$$

Extrémums (3)

Du fait du traitement du signe indépendamment par $(-1)^s$ la distribution des nombres flottants est symétrique par rapport à zéro on à :

$$x \in \mathbb{F} \Longleftrightarrow -x \in \mathbb{F}$$

$-x_{max}$ et $-x_{min}$ sont donc respectivement les plus petits et les plus grands nombres négatifs et normalisés. Dans le cas de l'utilisation de nombres dénormalisés ($denorm = vrai$), il y a aussi un plus petit nombre positif et dénormalisé

$$\bar{x}_{min} = b^{e_{min}-p}$$

ainsi qu'un plus grand nombre négatif et dénormalisé $-\bar{x}_{min}$ dans $\mathbb{F}$

Distance entre nombres flottants

Pour chaque choix de $e \in [e_{min}, e_{max}]$, les plus grandes et les plus petites valeurs de la mantisse des nombres flottants normalisés sont caractérisés respectivement par les valeurs

$$d_1 = 1,\ d_2 = \cdots = d_p = 0 \ \text{ et } \ d_1 = d_2 = \cdots = d_p = \delta := b - 1$$

on trouve alors pour les valeurs extrèmes de la mantisse

$$\begin{aligned} m_{min} &= .100\cdots 00_b = b^{-1} \\ m_{max} &= .\delta\delta\delta\cdots\delta\delta_b = \sum_{j=1}^{p}(b-1)b^{-j} = 1 - b^{-p} \end{aligned}$$

avec un incrément constant b^{-p}. Cet incrément basique est dénoté u.

Distance entre nombres flottants (2)

La distance entre deux nombre voisins de l'ensemble $\mathbb{F}_N$ dans l'intervalle $[b^e, b^{e+1}]$ est donc constante

$$\Delta x = b^{e-p} = ub^e$$

Donc chaque intervalle est caractérisé par une densité constante de nombres flottants normalisés.
Le passage d'un exposant e, à l'exposant suivant plus petit, réduit la distance constante par un facteur b et accroit la densité de nombres dans $\mathbb{F}_N$ par le même facteur.
De la même façon lors du passage à l'exposant suivant, plus grand, la distance est augmentée et la densité réduite par un facteur b. La structure consiste donc en la répétition de groupes de $(b-1)b^{p-1}$ nombres équidistants, chaque groupe étant une image du précédent par un facteur b.

Distance entre nombres flottants (3)

Autour de zéro

Dans le cas ou $denorm = faux$, il n'y a que deux valeurs de $\mathbb{F}_N$ dans l'intervalle $[0, x_{min}]$, les deux valeurs sur la frontière 0 et x_{min}.
Au contraire dans le cas ou $denorm = vrai$ alors l'intervalle $[0, x_{min}]$ est couvert uniformément par $b^{p-1} - 1$ nombres dénormalisés avec une distance constante $ub^{e_{min}}$ entre deux nombres voisins. Alors le plus petit nombre positif et dénormalisé

$$\bar{x} := \min\{x \in \mathbb{F}_D : x > 0\} = ub^{e_{min}} = b^{e_{min}-p}$$

est plus proche de zéro que le plus petit nombre positif normalisé $x_{min} = b^{e_{min}-1}$. Les nombres négatifs dans $\mathbb{F}_D$ sont obtenus par symétrie par rapport à l'origine.

Standardisation

Les standards internationaux sont en général développés par l'ISO (International Standardization Organization), mais le champ de l'électronique et de l'électricité est un cas particulier. les standards sont développés par l'IEC (International Electrotechnical Commission.
Dans les années 70, pour permettre la portabilité des prorgammes et rendre plus uniforme les problèmes d'arrondis et d'exceptions arithmétique, l'association Américaine IEEE (Institute of Electrical and Electronics Engineers) adopte un standard dénommé IEEE-754.
En 1989 l'IEC décide d'adopter ce dernier et de lui donner un statut international.

IEEE-754

Le fameux IEEE 754 spécifie :

- la description de deux classes de systèmes de calculs flottants, un format basique ainsi qu'un format étendu, chaque classe comprend des formats pour une précision simple et une précision double,
- les opérations élémentaires ainsi que les règles d'arrondi disponibles,
- Les conversions entre les différents formats de nombre ainsi qu'entre les nombres décimaux et binaires,
- le traitement des cas particuliers comme par exemple, le débordement des exposants ou les divisions par zéro.

IEEE-754 (2)

Les Formats basiques correspondent aux systèmes de nombres flottants, en simple précision $\mathbb{F}(2, 24, -125, 128, \textit{vrai})$ et à celui en double précision $\mathbb{F}(2, 53, -1021, 1024, \textit{vrai})$. Les formats étendus ne sont pas complètement définis par le standard qui impose seulement des bornes inférieures ou supérieures pour les paramètres.
Pour permettre de donner un résultat pour chaque opération sur l'ensemble du domaine $\mathbb{F}$ des nombres flottant, il a été ajouté des valeurs non numériques. Elles permettent de gérer correctement les opérations qui pourraient engendrer une erreur.

Des valeurs non numériques

- **+Inf, -Inf** $[+\infty, -\infty]$
  Ce sont deux valeurs infinies qui représentent à la fois un dépassement de capacité positif ou négatif et les résultats respectifs de $a/0^+$ et $a/0^-$ avec $a > 0$.
- **+Zéro, -Zéro** $[0^+, 0^-]$
  La présence de deux codages pour zéro est une conséquence des valeurs **+Inf** et **-Inf**.On applique la règle des signes pour ces deux valeurs on a donc $0^+ = (-1) * 0^-$. Il est nécessaire également que $0^+ = 0^-$ soit vrai malgré le fait que 0^+ et 0^- aient un codage différent.

Des valeurs non numériques (2)

- **Not a Number** [*NaN*]
  Cette valeur permet d'exprimer des résultats non exprimable dans le système comme $\sqrt{-1}$ ou qui ne peuvent être réduits, $0/0$ ou $\infty - \infty$. Lorsque la valeur *NaN* apparaît dans un calcul tous les calculs qui dépendent de cette variable ont pour résultat *NaN*.

Toutes les valeurs non numériques précédentes sont signalées par leur exposant comme le montre le tableau suivant.

Exposant	Fraction	Signification
$e_{max} + 1$	0	$\pm\infty$
$e_{max} + 1$	Non nulle	*NaN*
$-e_{max}$	Non nulle	Dénormalisé
$-e_{max}$	0	0

Additions non numériques

Le standart défini les règles d'additions avec des valeurs non numériques :

Somme	$-\infty$	$v \in \mathbb{F}^*_-$	0^-	0^+	$v \in \mathbb{F}^*_+$	$+\infty$	*NaN*
$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	*NaN*	*NaN*
$w \in \mathbb{F}^*_-$	$-\infty$	$v+w$ ou $-\infty$	w	w	$v+w$	$+\infty$	*NaN*
0_-	$-\infty$	v	0^-	0^+	v	$+\infty$	*NaN*
0_+	$-\infty$	v	0^+	0^+	v	$+\infty$	*NaN*
$w \in \mathbb{F}^*_+$	$-\infty$	$v+w$	w	w	$v+w$ ou $+\infty$	$+\infty$	*NaN*
$+\infty$	*NaN*	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	*NaN*

Multiplications non numériques

Les multiplications pour les valeurs non numériques :

Produit	$-\infty$	$v \in \mathbb{F}^*_-$	0^-	0^+	$v \in \mathbb{F}^*_+$	$+\infty$	*NaN*
$-\infty$	$+\infty$	$+\infty$	*NaN*	*NaN*	$-\infty$	$-\infty$	*NaN*
$w \in \mathbb{F}^*_-$	$+\infty$	$v * w$ ou $+\infty$	0^+	0^-	$v * w$ ou $-\infty$	$-\infty$	*NaN*
0_-	*NaN*	0^+	0^+	0^-	0^-	*NaN*	*NaN*
0_+	*NaN*	0^-	0^-	0^+	0^+	*NaN*	*NaN*
$w \in \mathbb{F}^*_+$	$-\infty$	$v * w$ ou $-\infty$	0^-	0^+	$v * w$ ou $+\infty$	$+\infty$	*NaN*
$+\infty$	$-\infty$	$-\infty$	*NaN*	*NaN*	$+\infty$	$+\infty$	*NaN*

les calculs

Le seul fait de ne faire les calculs que sur un ensemble fini de nombres, implique de nombreuses restrictions. Pour plus de simplicité on va supposer dans la suite n'effectuer les opérations qu'en utilisant un seul système de calcul flottant $\mathbb{F}(b, p, e_{min}, e_{max}, denorm)$.
Comme on a $\mathbb{F} \subset \mathbb{R}$, on appellera dans la suite le résultat *exact*, le résultat du calcul exécuté dans les réels, $\mathbb{R}$. Pour travailler dans $\mathbb{F}$, il faut donc arrondir le résultat exact pour qu'il appartienne à $\mathbb{F}$.

Une projection de $\mathbb{R}$ sur $\mathbb{F}$

Une fonction d'arrondi est une projection

$$\square : \mathbb{R} \to \mathbb{F}$$

qui associe à chaque réel x, un nombre $\square x \in \mathbb{F}$.
Etant donné une fonction d'arrondi $\square$, on peut construire une définition mathématique de l'arithmétique dans $\mathbb{F}$.
Pour chaque opération arithmétique dans $\mathbb{R}$

$$\circ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

Dans $\mathbb{F}$

l'opération correspondante dans $\mathbb{F}$

$$\boxdot : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$$

peut être définie par

$$x \boxdot y := \square(x \circ y)$$

Le modèle consiste en deux étapes, on détermine d'abord le résultat exact $x \circ y$ de l'opération $\circ$, puis on projette le résultat dans $\mathbb{F}$ en utilisant la fonction d'arrondi $\square$, on a bien alors $x \boxdot y$ dans $\mathbb{F}$.

Opérations unaires

On peut construire de la même façon les opérations unaires. Pour une fonction $f : \mathbb{R} \to \mathbb{R}$ la fonction correspondante dans $\mathbb{F}$ est

$$\boxed{f} : \mathbb{F} \to \mathbb{F}$$

définie par

$$\boxed{f}(x) := \square f(x)$$

Bien entendu comme les fonctions d'arrondis sont des projections, elles en possèdent les propriétés, on a donc

$$\square x = x \quad \forall x \in \mathbb{F}$$

Monotonie

De plus elles sont monotones

$$x \leq y \implies \square x \leq \square y \quad \forall \; x, y \in \mathbb{R}$$

ces deux propriétés, impliquent, qu'une telle fonction, arrondie tous les nombres réels, compris entre deux nombres x_1 et x_2 voisins dans $\mathbb{F}$, par x_1, lorsque le réel est plus petit qu'un point limite $\hat{x} \in [x_1, x_2]$ et par x_2, lorsqu'il est plus grand. Si $\hat{x}$ n'est égal ni à x_1 ni à x_2 il faut donc spécifier comment calculer $\hat{x}$, pour pouvoir trouver $\square\hat{x} = x_1$ ou $\square\hat{x} = x_2$.

Arrondi au plus proche

Arrondi au plus proche (arrondi optimal)

Cette fonction fixe le point limite au milieu de x_1 et x_2 :

$$\hat{x} := \frac{x_1 + x_2}{2}$$

Dans ce cas $x \in \mathbb{R}_N \cup \mathbb{R}_D$ est toujours arrondi au nombre le plus proche dans $\mathbb{F}$. Pour $x = \hat{x}$ équidistant de x_1 et x_2, il y a deux solutions :

On peut choisir le nombre $\square x$ le plus loin de zéro, ou bien décider en fonction du dernier digit de la mantisse.

Arrondi vers zéro

Le point limite dépend du signe du nombre x :

$$\hat{x} := sign(x) \max(|x_1|, |x_2|)$$

où $sign(x) := -1$ pour $x < 0$ et $sign(x) := 1$ pour $x \geq 0$ dans ce cas le point limite $\hat{x}$ est celui des deux voisins x_1 et x_2, qui à la plus grande valeur absolue, autrement dit pour $x \notin \mathbb{F}$, $\square x$ est toujours le nombre parmi x_1 et x_2, ayant la plus petite valeur absolue et

$$x \in \mathbb{R} \backslash \mathbb{F} \Rightarrow |\square x| < |x|$$

Il est bien entendu parfaitement possible de prendre au contraire un arrondi s'éloignant de zéro, mais ce n'est pas usuel.

Arrondi direct

Pour *l'arrondi vers plus l'infini* et *l'arrondi vers moins l'infini*, les points limites sont donnés par

$$\hat{x} := \min(x_1, x_2) \quad \text{et} \quad \hat{x} := \max(x_1, x_2)$$

Le résultat de l'arrondi est alors le nombre suivant, dans l'ordre décroissant de $\mathbb{F}$, ou dans l'ordre croissant, quelque soit le signe de $x \notin \mathbb{F}$. On a alors si $x \notin \mathbb{F}$ $\square x > x$ et $\square x < x$ respectivement.
La relation

$$\square(-x) = -(\square x)$$

est toujours vraie dans les cas d'arrondi optimal ou de troncature, ces deux fonctions d'arrondi sont alors appelées fonctions d'arrondi symétrique par opposition à la fonction d'arrondi directe.

Erreur absolue

La différence entre l'arrondi $\square x \in \mathbb{F}$ et la valeur exacte $x \in \mathbb{R}$ est appelée *l'erreur absolue d'arrondi* de x :

$$\epsilon(x) := \square x - x$$

Erreur relative

alors que

$$\rho(x) := \frac{\square x - x}{x} = \frac{\epsilon(x)}{x}$$

est *l'erreur relative d'arrondi* de x.

Bornes supérieure de l'erreur absolue

La valeur absolue $|\epsilon(x)|$ est limité par Δx, la longueur du plus petit intervalle $[x_1, x_2]$ qui contient x, avec $x_1, x_2 \in \mathbb{F}$ et dans le cas d'arrondi au plus proche cette longueur divisée par deux.
pour chaque $x \in \mathbb{R}_N$ la valeur $\square x \in \mathbb{F}$ admet une représentation unique

$$\square x = (-1)^s m(x) b^{e(x)}$$

où $m(x)$ est la mantisse et $e(x)$ l'exposant de $\square x$.

Bornes supérieure de l'erreur absolue (2)

Comme la longueur du plus petit intervalle $[x_1, x_2]$ avec $x_1, x_2 \in \mathbb{F}_N$ qui contient $x \in \mathbb{R}_N \backslash \mathbb{F}_N$ est

$$\Delta x = ub^{e(x)}$$

l'erreur absolue d'arrondi de n'importe quel $x \in \mathbb{R}_N$ est

$$|\epsilon(x)| \begin{cases} < & ub^{e(x)} \text{ pour l'arrondi direct ou la troncature} \\ \leq & \frac{u}{2} b^{e(x)} \text{ pour l'arrondi au plus proche} \end{cases}$$

Pour $x \in \mathbb{R}_D$ et l'arrondi des nombres dénormalisés l'exposant $e(x)$ est remplaçé par e_{min}.

Bornes supérieures de l'erreur relative

Pour l'erreur d'arrondi relative $\rho(x)$, il y a des bornes supérieures pour l'ensemble de $\mathbb{R}_N$.
Les plus petites bornes uniformes sont

$$|\rho(x)| \begin{cases} < & \frac{u}{|m(x)|} \leq bu \quad \text{pour l'arrondi direct et la troncature} \\ \leq & \frac{u}{2|m(x)|} \leq bu/2 \quad \text{pour l'arrondi au plus proche} \end{cases}$$

On appelle en général *eps* la borne supérieure pour l'erreur relative

Bornes supérieures de l'erreur relative (2)

$$eps = \begin{cases} bu & = b^{1-p} \quad \text{pour l'arrondi direct et la troncature} \\ bu/2 & = b^{1-p}/2 \quad \text{pour l'arrondi au plus proche} \end{cases}$$

On peut montrer le théorème suivant :

Theorem

Pour chaque $x \in \mathbb{R}_N$ il existe un $\rho \in \mathbb{R}$ avec

$$\square x = x(1+\rho) \quad et \quad |\rho| \leq eps$$

Attention

Lorsque l'on utilise la borne uniforme

$$|\rho(x)| \leq \mathit{eps} \quad \text{pour} \quad x \in \mathbb{R}_N$$

il faut se rappeler que c'est une estimation qui peut être trés pessimiste. Si l'on a plus d'information sur la position de la mantisse $m(x)$ dans l'intervalle $[b^{-1}, 1 - b^{-p}]$, on peut utiliser plutôt

$$|\rho(x)| \leq \frac{u}{m(x)} \quad \text{ou} \quad |\rho(x)| \leq \frac{u}{2m(x)}$$

Une conséquence directe du théorème 1 en utilisant les définitions utilisées précedemment nous donne

Theorem

A condition que le résultat exact d'une opération arithmétique $\circ$ soit dans $\mathbb{R}_N$, alors pour tous les $x, y \in \mathbb{F}_N$, il existe un $\rho \in \mathbb{R}$, tel que

$$x \boxdot y = (x \circ y)(1 + \rho) \quad et \quad |\rho| \leq eps$$

Theorem

A condition que pour l'argument $x \in D \cap \mathbb{F}_N$, la valeur exacte de $f : D \to \mathbb{R}$ soit dans $\mathbb{R}_N$, il existe un $\rho \in \mathbb{R}$, tel que

$$\boxed{f}(x) = f(x)(1 + \rho) \quad et \quad |\rho| \leq eps$$

Conséquence

Une conséquence importante est que de manière générale

$$x \boxplus (y \boxplus z) \neq (x \boxplus y) \boxplus z$$

de même

$$x \boxdot (y \boxdot z) \neq (x \boxdot y) \boxdot z$$

Donc l'associativité de la multiplication et de l'addition des nombres réels n'est pas conservé par les systèmes de calculs flottants, dans $\mathbb{F}$. La distributivité de la multiplication sur l'addition n'est pas non plus valide dans $\mathbb{F}$

$$x \boxdot (y \boxplus z) \neq (x \boxdot y) \boxplus (x \boxdot z)$$

Mais

$$x \boxplus y = \square(x + y) = \square(y + x) = y \boxplus x$$

l'addition et la multiplication restent commutative dans $\mathbb{F}$.

fonctions

L'entète cmath de la librairie standard comprend un certain nombre de fonctions utiles (on suppose utiliser $C++11$), toutes les fonctions sont surchargée pour être utilisées avec *float* et *long double*.

```
int ilogb (double x);
double frexp (double x, int* exp);
double ldexp (double x, int exp);
double scalbn (double x, int n);
bool signbit (double x);
double exp (double x);
double exp2 (double x);
double modf (double x, double* intpart);
```

(http ://www.cplusplus.com/reference/cmath)