ANALYSE NUMÉRIQUE I

COURS 2

STANDARD IEEE 754

Ch. Baskiotis

LAPI – EISTI

29 novembre 2008

Standard IEEE 754

Pour les nombres flottants en simple précision le standard IEEE 754 est

Signe s	Exposant e	Mantisse m
s = 1 bit	q = 8 bits	p = 23 bits
□	□□□□□□□□	□□□□□□□□□□□□□□□□□□□□□□□
$\pm$	$a_1a_2a_3a_4a_5a_6a_7a_8$	$b_1b_2b_3b_4b_5b_6b_7b_8b_9b_{10}b_{11}b_{12}b_{13}b_{14}b_{15}b_{16}b_{17}b_{18}b_{19}b_{20}b_{21}b_{22}b_{23}$

qui est un nombre sur 32 bits.

À l'intérieur de la cuisine IEEE 754 – I

Bits de l'exposant $a_1 \cdots a_8$	La valeur numérique correspondante
$(00000000)_2 = 0$	$\pm\,(0.b_1 b_2 \cdots b_{23})_2 \times 2^{-127}$
$(00000001)_2 = 1$	$\pm\,(1.b_1 b_2 \cdots b_{23})_2 \times 2^{-126}$
$(00000010)_2 = 2$	$\pm\,(1.b_1 b_2 \cdots b_{23})_2 \times 2^{-125}$
$\vdots$	$\vdots$

À l'intérieur de la cuisine IEEE 754 – II

Bits de l'exposant $a_1 \cdots a_8$	La valeur numérique correspondante
$\vdots$	$\vdots$
$(01111111)_2 = 127$	$\pm(1.b_1b_2\cdots b_{23})_2 \times 2^0$
$(10000000)_2 = 128$	$\pm(1.b_1b_2\cdots b_{23})_2 \times 2^1$
$\vdots$	$\vdots$

À l'intérieur de la cuisine IEEE 754 – III

Bits de l'exposant $a_1 \cdots a_8$	La valeur numérique correspondante
$\vdots$	$\vdots$
$(11111101)_2 = 253$	$\pm (1.b_1 b_2 \cdots b_{23})_2 \times 2^{126}$
$(11111110)_2 = 254$	$\pm (1.b_1 b_2 \cdots b_{23})_2 \times 2^{127}$
$(11111111)_2 = 255$	$\pm\infty$ si $b_1 = b_2 = \cdots = b_{23} = 0$, NaN sinon.

De zéro à l'infini en passant par le plus petit et le plus grand positifs – I

Le zéro

0	00000000	00000000000000000000000

De zéro à l'infini en passant par le plus petit et le plus grand positifs – I

Le zéro

0	00000000	00000000000000000000000

et le plus petit positif

0	00000001	00000000000000000000000

qui est égal à 1×2^{-126}.

De zéro à l’infini en passant par le plus petit et le plus grand positifs – II

Le plus grand positif

0	11111110	11111111111111111111111

ce qui équivaut à $(1.11\ldots1)_2 \times 2^{127} = \left(2 - 2^{-23}\right) \times 2^{127} \approx 3.4 \times 10^{38}$

De zéro à l'infini en passant par le plus petit et le plus grand positifs – II

Le plus grand positif

0	11111110	11111111111111111111111

ce qui équivaut à $(1.11\ldots1)_2 \times 2^{127} = \left(2 - 2^{-23}\right) \times 2^{127} \approx 3.4 \times 10^{38}$

et... l'infini

0	11111111	00000000000000000000000

Nombres sous-normalisés

Tout nombre codé de la façon suivante

0	00000000	$XXXXXXXXXXXXXXXXXXXXXXX$

avec $X = 0$ ou 1 et au moins une valeur différente de 0,
est un nombre ***sous-normalisé***.

Nombres sous-normalisés

Tout nombre codé de la façon suivante

0	00000000	$XXXXXXXXXXXXXXXXXXXXXXX$

avec $X = 0$ ou 1 et au moins une valeur différente de 0,
est un nombre ***sous-normalisé***.

$\Longrightarrow$ L'ordinateur peut faire des calculs avec ces nombres

Nombres sous-normalisés

Tout nombre codé de la façon suivante

0	00000000	$XXXXXXXXXXXXXXXXXXXXXXX$

avec $X = 0$ ou 1 et au moins une valeur différente de 0,
est un nombre ***sous-normalisé***.

$\Longrightarrow$ L'ordinateur peut faire des calculs avec ces nombres
mais l'utilisateur ne peut les manipuler.

Nombres sous-normalisés

Tout nombre codé de la façon suivante

0	00000000	$XXXXXXXXXXXXXXXXXXXXXXX$

avec $X = 0$ ou 1 et au moins une valeur différente de 0,
est un nombre ***sous-normalisé***.

$\Rightarrow$ L'ordinateur peut faire des calculs avec ces nombres mais l'utilisateur ne peut les manipuler.

$\Rightarrow$ La précision est mauvaise.

Les Not-a-Number (NaN

Tout nombre codé de la façon suivante

0	11111111	$XXXXXXXXXXXXXXXXXXXXXXX$

avec $X = 0$ ou 1 et au moins une valeur différente de 0,
est un ***Not-a-Number (NaN)***.

Les Not-a-Number (NaN

Tout nombre codé de la façon suivante

0	11111111	$XXXXXXXXXXXXXXXXXXXXXXX$

avec $X = 0$ ou 1 et au moins une valeur différente de 0,
est un ***Not-a-Number (NaN)***.

$\Rightarrow$ L'ordinateur refuse de faire des calculs avec ces nombres.

Du côté de l'exposant - I
On se signe ou pas

Les valeurs de l'exposant sont des entiers, positifs ou négatifs. Mais l'exposant va de 0 à $2^q - 1$.

Doit-on tenir compte du signe ?

Du côté de l'exposant - I
On se signe ou pas

Les valeurs de l'exposant sont des entiers, positifs ou négatifs. Mais l'exposant va de 0 à $2^q - 1$.

Doit-on tenir compte du signe ?
$\Rightarrow$ Réponse : oui.

Comment faire ? Peut-être en réservant un bit pour le signe de l'exposant.

Du côté de l'exposant - I
On se signe ou pas

Les valeurs de l'exposant sont des entiers, positifs ou négatifs. Mais l'exposant va de 0 à $2^q - 1$.

Doit-on tenir compte du signe ?
$\Rightarrow$ Réponse : oui.

Comment faire ? Peut-être en réservant un bit pour le signe de l'exposant.
$\Rightarrow$ Réponse : non, car trop compliqué.

Du côté de l'exposant - II
On se signe mais différemment

On coupe l'intervalle des naturels $[0,\, 2^q - 1]$ en deux parties égales.

Du côté de l'exposant - II
On se signe mais différemment

On coupe l'intervalle des naturels $[0, 2^q - 1]$ en deux parties égales.

La valeur du milieu, $B = 2^{q-1} - 1$ sera le 0 de l'exposant.

Du côté de l'exposant - II
On se signe mais différemment

On coupe l'intervalle des naturels $[0, 2^q - 1]$ en deux parties égales.

La valeur du milieu, $B = 2^{q-1} - 1$ sera le 0 de l'exposant.

Par convention
– Toute valeur inférieure à B sera une valeur négative pour l'exposant.

Du côté de l'exposant - II
On se signe mais différemment

On coupe l'intervalle des naturels $[0, 2^q - 1]$ en deux parties égales.

La valeur du milieu, $B = 2^{q-1} - 1$ sera le 0 de l'exposant.

Par convention
- Toute valeur inférieure à B sera une valeur négative pour l'exposant.
- Toute valeur supérieure à B sera une valeur positive pour l'exposant.

Du côté de l'exposant - II
On se signe mais différemment

On coupe l'intervalle des naturels $[0, 2^q - 1]$ en deux parties égales.

La valeur du milieu, $B = 2^{q-1} - 1$ sera le 0 de l'exposant.

Par convention
- Toute valeur inférieure à B sera une valeur négative pour l'exposant.
- Toute valeur supérieure à B sera une valeur positive pour l'exposant.

$\Rightarrow$ Si pour un nombre, la valeur de son exposant est e,

Du côté de l'exposant - II
On se signe mais différemment

On coupe l'intervalle des naturels $[0, 2^q - 1]$ en deux parties égales.

La valeur du milieu, $B = 2^{q-1} - 1$ sera le 0 de l'exposant.

Par convention
- Toute valeur inférieure à B sera une valeur négative pour l'exposant.
- Toute valeur supérieure à B sera une valeur positive pour l'exposant.

$\Rightarrow$ Si pour un nombre, la valeur de son exposant est e,
le codage de cet exposant aura la valeur $E = e + B$

Du côté de l'exposant - II
On se signe mais différemment

On coupe l'intervalle des naturels $[0, 2^q - 1]$ en deux parties égales.

La valeur du milieu, $B = 2^{q-1} - 1$ sera le 0 de l'exposant.

Par convention
- Toute valeur inférieure à B sera une valeur négative pour l'exposant.
- Toute valeur supérieure à B sera une valeur positive pour l'exposant.

$\Rightarrow$ Si pour un nombre, la valeur de son exposant est e,
le codage de cet exposant aura la valeur $E = e + B$
$\Rightarrow$ et au décodage on aura $e = E - B$.

Du côté de l'exposant - II
On se signe mais différemment

On coupe l'intervalle des naturels $[0,\, 2^q - 1]$ en deux parties égales.

La valeur du milieu, $B = 2^{q-1} - 1$ sera le 0 de l'exposant.

Par convention
- Toute valeur inférieure à B sera une valeur négative pour l'exposant.
- Toute valeur supérieure à B sera une valeur positive pour l'exposant.

$\Rightarrow$ Si pour un nombre, la valeur de son exposant est e,
le codage de cet exposant aura la valeur $E = e + B$
$\Rightarrow$ et au décodage on aura $e = E - B$.

La valeur B est appelée le ***biais du codage***.

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 \, = \, 2^7 - 1 \, = \, 127$.

Du côté de l'exposant - III

Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

– Exposant du nombre e $=$ 0 $\Rightarrow$ Exposant du code E $=$ 127.

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

- Exposant du nombre e $= 0 \Rightarrow$ Exposant du code E $= 127$.
- Exposant du nombre e $< 0 \Rightarrow$ Exposant du code $1 \leq$ E < 127.

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

- Exposant du nombre e $=$ 0 $\Rightarrow$ Exposant du code E $=$ 127.
- Exposant du nombre e $<$ 0 $\Rightarrow$ Exposant du code 1 $\leq$ E $<$ 127.
- Exposant du nombre e $>$ 0 $\Rightarrow$ Exposant du code 127 $<$E $\leq$ 254.

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

- Exposant du nombre e $= 0 \Rightarrow$ Exposant du code E $= 127$.
- Exposant du nombre e $< 0 \Rightarrow$ Exposant du code $1 \leq$ E < 127.
- Exposant du nombre e $> 0 \Rightarrow$ Exposant du code 127 <E $\leq$ 254.

Exemple : $5.75_{10} = 101.11_2 = 1.0111 \times 10^{10}\,_2 \Rightarrow$

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

- Exposant du nombre e = 0 $\Rightarrow$ Exposant du code E = 127.
- Exposant du nombre e < 0 $\Rightarrow$ Exposant du code 1 $\leq$ E < 127.
- Exposant du nombre e > 0 $\Rightarrow$ Exposant du code 127 <E $\leq$ 254.

Exemple : $5.75_{10} = 101.11_2 = 1.0111 \times 10^{10}\ _2 \Rightarrow$

Donc exposant $e = 10_2 = 2_{10}$

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

– Exposant du nombre e = 0 ⇒ Exposant du code E = 127.

– Exposant du nombre e < 0 ⇒ Exposant du code 1 ≤ E < 127.

– Exposant du nombre e > 0 ⇒ Exposant du code 127 <E ≤ 254.

Exemple : $5.75_{10} = 101.11_2 = 1.0111 \times 10^{10}\ _2 \Rightarrow$

Donc exposant $e = 10_2 = 2_{10}$ et donc exposant codé

$E = e + B = 127 + 2 = 129_{10} = 100000001_2$

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

- Exposant du nombre e = 0 $\Rightarrow$ Exposant du code E = 127.
- Exposant du nombre e < 0 $\Rightarrow$ Exposant du code 1 $\leq$ E < 127.
- Exposant du nombre e > 0 $\Rightarrow$ Exposant du code 127 <E $\leq$ 254.

Exemple : $5.75_{10} = 101.11_2 = 1.0111 \times 10^{10}{}_2 \Rightarrow$

Donc exposant $e = 10_2 = 2_{10}$ et donc exposant codé

$E = e + B = 127 + 2 = 129_{10} = 100000001_2 \Rightarrow$Donc le nombre est codé

0	10000001	01110000000000000000000

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

- Exposant du nombre e = 0 $\Rightarrow$ Exposant du code E = 127.
- Exposant du nombre e < 0 $\Rightarrow$ Exposant du code 1 $\leq$ E < 127.
- Exposant du nombre e > 0 $\Rightarrow$ Exposant du code 127 <E $\leq$ 254.

Exemple : $5.75_{10} = 101.11_2 = 1.0111 \times 10^{10}\ {}_2 \Rightarrow$

Donc exposant $e = 10_2 = 2_{10}$ et donc exposant codé

$E = e + B = 127 + 2 = 129_{10} = 100000001_2 \Rightarrow$Donc le nombre est codé

0	10000001	01110000000000000000000

Exemple : $0.75_{10} = 0.11_2 = 1.1 \times 10^{-1}\ {}_2 \Rightarrow$

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

- Exposant du nombre e = 0 ⇒ Exposant du code E = 127.
- Exposant du nombre e < 0 ⇒ Exposant du code 1 ≤ E < 127.
- Exposant du nombre e > 0 ⇒ Exposant du code 127 <E ≤ 254.

Exemple : $5.75_{10} = 101.11_2 = 1.0111 \times 10^{10}\ {}_2 \Rightarrow$

Donc exposant $e = 10_2 = 2_{10}$ et donc exposant codé

$E = e + B = 127 + 2 = 129_{10} = 100000001_2 \Rightarrow$Donc le nombre est codé

0	10000001	01110000000000000000000

Exemple : $0.75_{10} = 0.11_2 = 1.1 \times 10^{-1}\ {}_2 \Rightarrow$

Donc exposant $e = -1_2 = -1_{10}$

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

- Exposant du nombre e = 0 $\Rightarrow$ Exposant du code E = 127.
- Exposant du nombre e < 0 $\Rightarrow$ Exposant du code 1 $\leq$ E < 127.
- Exposant du nombre e > 0 $\Rightarrow$ Exposant du code 127 <E $\leq$ 254.

Exemple : $5.75_{10} = 101.11_2 = 1.0111 \times 10^{10}\ _2 \Rightarrow$

Donc exposant $e = 10_2 = 2_{10}$ et donc exposant codé

$E = e + B = 127 + 2 = 129_{10} = 100000001_2 \Rightarrow$Donc le nombre est codé

0	10000001	01110000000000000000000

Exemple : $0.75_{10} = 0.11_2 = 1.1 \times 10^{-1}\ _2 \Rightarrow$

Donc exposant $e = -1_2 = -1_{10}$ et donc exposant codé

$E = e + B = 127 - 1 = 126_{10} = 01111110_2$

Du côté de l'exposant - III
Application au standard IEEE-754

$q = 8 \Rightarrow$ Biais $B = 2^{q-1} - 1 = 2^7 - 1 = 127$.

- Exposant du nombre e = 0 $\Rightarrow$ Exposant du code E = 127.
- Exposant du nombre e < 0 $\Rightarrow$ Exposant du code 1 $\leq$ E < 127.
- Exposant du nombre e > 0 $\Rightarrow$ Exposant du code 127 <E $\leq$ 254.

Exemple : $5.75_{10} = 101.11_2 = 1.0111 \times 10^{10}\ _2 \Rightarrow$

Donc exposant $e = 10_2 = 2_{10}$ et donc exposant codé

$E = e + B = 127 + 2 = 129_{10} = 10000001_2 \Rightarrow$Donc le nombre est codé

0	10000001	01110000000000000000000

Exemple : $0.75_{10} = 0.11_2 = 1.1 \times 10^{-1}\ _2 \Rightarrow$

Donc exposant $e = -1_2 = -1_{10}$ et donc exposant codé

$E = e + B = 127 - 1 = 126_{10} = 01111110_2 \Rightarrow$Donc codage du nombre

0	01111110	10000000000000000000000

Laboratoire EISTI LAπPI en Processus Intelligents

Le miracle est (presque) impossible

Le stockage d'un nombre dans un ordinateur,
produit une erreur sur la valeur du nombre.

Le miracle est (presque) impossible

Le stockage d'un nombre dans un ordinateur,
produit une erreur sur la valeur du nombre.
Donc le stockage de deux nombres dans un ordinateur, produit deux erreurs.

Le miracle est (presque) impossible

Le stockage d'un nombre dans un ordinateur,
produit une erreur sur la valeur du nombre.
Donc le stockage de deux nombres dans un ordinateur, produit deux erreurs.

Si on fait, avec ces deux nombres, une opération, on produit une 3e erreur

Le miracle est (presque) impossible

Le stockage d'un nombre dans un ordinateur,
produit une erreur sur la valeur du nombre.
Donc le stockage de deux nombres dans un ordinateur, produit deux erreurs.

Si on fait, avec ces deux nombres, une opération, on produit une 3e erreur, sauf à espérer que les erreurs se compensent

Le miracle est (presque) impossible

Le stockage d'un nombre dans un ordinateur,
produit une erreur sur la valeur du nombre.
Donc le stockage de deux nombres dans un ordinateur, produit deux erreurs.

Si on fait, avec ces deux nombres, une opération, on produit une 3e erreur,
sauf à espérer que les erreurs se compensent et
le nombre sort de la bassine de calcul exempt, comme par miracle, de tout défaut.

Le miracle est (presque) impossible

Le stockage d'un nombre dans un ordinateur,
produit une erreur sur la valeur du nombre.
Donc le stockage de deux nombres dans un ordinateur, produit deux erreurs.

Si on fait, avec ces deux nombres, une opération, on produit une 3e erreur,
sauf à espérer que les erreurs se compensent et
le nombre sort de la bassine de calcul exempt, comme par miracle, de tout défaut.

$\Rightarrow$ En analyse numérique, il n'y pas de miracles.

Le miracle est (presque) impossible

Le stockage d'un nombre dans un ordinateur,
produit une erreur sur la valeur du nombre.
Donc le stockage de deux nombres dans un ordinateur, produit deux erreurs.

Si on fait, avec ces deux nombres, une opération, on produit une 3e erreur,
sauf à espérer que les erreurs se compensent et
le nombre sort de la bassine de calcul exempt, comme par miracle, de tout défaut.

$\Rightarrow$ En analyse numérique, il n'y pas de miracles.
Depuis le dernier conseil pédagogique, j'ai révisé à la baisse les objectifs du cours.

Le miracle est (presque) impossible

Le stockage d'un nombre dans un ordinateur,
produit une erreur sur la valeur du nombre.
Donc le stockage de deux nombres dans un ordinateur, produit deux erreurs.

Si on fait, avec ces deux nombres, une opération, on produit une 3e erreur,
sauf à espérer que les erreurs se compensent et
le nombre sort de la bassine de calcul exempt, comme par miracle, de tout défaut.

$\Rightarrow$ En analyse numérique, il n'y pas de miracles.
Depuis le dernier conseil pédagogique, j'ai révisé à la baisse les objectifs du cours.

L'objectif du cours est d'abolir la pensée magique
(tout au plus vous pouvez garder la pensée sauvage $\Rightarrow$ cf.C.L.-S.)

Le miracle est (presque) impossible

Le stockage d'un nombre dans un ordinateur,
produit une erreur sur la valeur du nombre.
Donc le stockage de deux nombres dans un ordinateur, produit deux erreurs.

Si on fait, avec ces deux nombres, une opération, on produit une 3e erreur, sauf à espérer que les erreurs se compensent et le nombre sort de la bassine de calcul exempt, comme par miracle, de tout défaut.

$\Rightarrow$ En analyse numérique, il n'y pas de miracles.
Depuis le dernier conseil pédagogique, j'ai révisé à la baisse les objectifs du cours.

L'objectif du cours est d'abolir la pensée magique
(tout au plus vous pouvez garder la pensée sauvage $\Rightarrow$ cf.C.L.-S.)

L'analyse numérique vise à faire comprendre à l'élève que les miracles en calcul numérique sont (presque) impossibles.

Opérations arithmétiques et erreur

$a, b \in \mathbb{R}$ et $\otimes$ opération arithmétique (c-à-d. $+$, $-$, $\times$, $/$, $\sqrt{}$).

Opérations arithmétiques et erreur

$a, b \in \mathbb{R}$ et $\otimes$ opération arithmétique (c-à-d. $+$, $-$, $\times$, $/$, $\sqrt{}$).

$c = a \otimes b$ où c est le résultat de l'opération.

Opérations arithmétiques et erreur

$a, b \in \mathbb{R}$ et $\otimes$ opération arithmétique (c-à-d. $+$, $-$, $\times$, $/$, $\sqrt{}$).

$c = a \otimes b$ où c est le résultat de l'opération.

Sur un ordinateur, on a :

Opérations arithmétiques et erreur

$a, b \in \mathbb{R}$ et $\otimes$ opération arithmétique (c-à-d. $+$, $-$, $\times$, $/$, $\sqrt{}$).

$c = a \otimes b$ où c est le résultat de l'opération.

Sur un ordinateur, on a :

$$m(c) = m(m(a) \otimes m(b))$$

Opérations arithmétiques et erreur

$a, b \in \mathbb{R}$ et $\otimes$ opération arithmétique (c-à-d. $+$, $-$, $\times$, $/$, $\sqrt{}$).

$c = a \otimes b$ où c est le résultat de l'opération.

Sur un ordinateur, on a :

$$m(c) = m(m(a) \otimes m(b))$$

L'erreur de précision $\eta(c) = \frac{m(c)-c}{c}$ peut se décomposer en deux parties :

Opérations arithmétiques et erreur

$a, b \in \mathbb{R}$ et $\otimes$ opération arithmétique (c-à-d. $+$, $-$, $\times$, $/$, $\sqrt{}$).
$c = a \otimes b$ où c est le résultat de l'opération.

Sur un ordinateur, on a :

$$m(c) = m(m(a) \otimes m(b))$$

L'erreur de précision $\eta(c) = \frac{m(c)-c}{c}$ peut se décomposer en deux parties :

1. ***une erreur du calcul***, notée $\eta^C(c) = \eta^C(a \otimes b)$, qui est caractéristique de la machine et contre laquelle on ne peut rien faire (sauf à changer de machine),

Opérations arithmétiques et erreur

$a, b \in \mathbb{R}$ et $\otimes$ opération arithmétique (c-à-d. $+$, $-$, $\times$, $/$, $\sqrt{}$).
$c = a \otimes b$ où c est le résultat de l'opération.
Sur un ordinateur, on a :

$$m(c) = m(m(a) \otimes m(b))$$

L'erreur de précision $\eta(c) = \frac{m(c)-c}{c}$ peut se décomposer en deux parties :

1. ***une erreur du calcul***, notée $\eta^C(c) = \eta^C(a \otimes b)$, qui est caractéristique de la machine et contre laquelle on ne peut rien faire (sauf à changer de machine), et
2. une ***erreur de l'entrée***, notée $\eta^I(c) = \eta^I(a \otimes b)$, due à la représentation par l'ordinateur du résultat de l'opération et qui dépend de l'ordre des calculs.

Opérations arithmétiques et erreur

$a, b \in \mathbb{R}$ et $\otimes$ opération arithmétique (c-à-d. $+$, $-$, $\times$, $/$, $\sqrt{}$).
$c = a \otimes b$ où c est le résultat de l'opération.
Sur un ordinateur, on a :

$$m(c) = m(m(a) \otimes m(b))$$

L'erreur de précision $\eta(c) = \frac{m(c)-c}{c}$ peut se décomposer en deux parties :

1. ***une erreur du calcul***, notée $\eta^C(c) = \eta^C(a \otimes b)$, qui est caractéristique de la machine et contre laquelle on ne peut rien faire (sauf à changer de machine), et
2. une ***erreur de l'entrée***, notée $\eta^I(c) = \eta^I(a \otimes b)$, due à la représentation par l'ordinateur du résultat de l'opération et qui dépend de l'ordre des calculs.
   $\Rightarrow$ **On change l'ordre de calculs** $\Rightarrow$ **on modifie l'erreur**.

Opérations arithmétiques et erreur – Addition

Théorème de l'Erreur pour les Sommes .- Soit la somme

$$S = x_1 + x_2 + \cdots$$

Opérations arithmétiques et erreur – Addition

THÉORÈME DE L'ERREUR POUR LES SOMMES .- Soit la somme

$$S = x_1 + x_2 + \cdots$$

L'erreur absolue de l'entrée pour cette somme est

$$\Delta^I S = \Delta^I x_1 + \Delta^I x_2 + \cdots$$

Opérations arithmétiques et erreur – Addition

THÉORÈME DE L'ERREUR POUR LES SOMMES .- Soit la somme

$$S = x_1 + x_2 + \cdots$$

L'erreur absolue de l'entrée pour cette somme est

$$\Delta^I S = \Delta^I x_1 + \Delta^I x_2 + \cdots$$

et l'erreur de précision de l'entrée est

$$\eta^I(S) = \frac{x_1}{S}\eta^I(x_1) + \frac{x_2}{S}\eta^I(x_2) + \cdots,$$ avec $\mid \eta^I(x_1) \mid \leq$ eps.

Opérations arithmétiques et erreur – Addition

THÉORÈME DE L'ERREUR POUR LES SOMMES .- Soit la somme

$$S = x_1 + x_2 + \cdots$$

L'erreur absolue de l'entrée pour cette somme est

$$\Delta^I S = \Delta^I x_1 + \Delta^I x_2 + \cdots$$

et l'erreur de précision de l'entrée est

$$\eta^I(S) = \frac{x_1}{S}\eta^I(x_1) + \frac{x_2}{S}\eta^I(x_2) + \cdots, \text{ avec } |\eta^I(x_1)| \leq \text{eps.}$$

$\Rightarrow$ Erreur maximale de la somme

$$|\eta^I(S)| \leq \left|\frac{x_1}{S}\right| |\eta^I(x_1)| + \left|\frac{x_2}{S}\right| |\eta^I(x_2)| + \cdots \leq \frac{1}{|S|}(|x_1| + |x_2| + \cdots) \cdot \text{eps}$$

Opérations arithmétiques et erreur – Addition

THÉORÈME DE L'ERREUR POUR LES SOMMES .- Soit la somme

$$S = x_1 + x_2 + \cdots$$

L'erreur absolue de l'entrée pour cette somme est

$$\Delta^I S = \Delta^I x_1 + \Delta^I x_2 + \cdots$$

et l'erreur de précision de l'entrée est

$$\eta^I\left(S\right) = \frac{x_1}{S}\eta^I\left(x_1\right) + \frac{x_2}{S}\eta^I\left(x_2\right) + \cdots \text{, avec } \mid \eta^I\left(x_1\right) \mid \leq \text{eps.}$$

$\Rightarrow$ Erreur maximale de la somme

$$\mid \eta^I\left(S\right) \mid \leq \mid \frac{x_1}{S} \mid\mid \eta^I\left(x_1\right) \mid + \mid \frac{x_2}{S} \mid\mid \eta^I\left(x_2\right) \mid + \cdots \leq \frac{1}{|S|}\left(\mid x_1 \mid + \mid x_2 \mid + \cdots\right) \cdot \text{eps}$$

$$\Rightarrow \ \mid \eta^I\left(S\right) \mid \leq \frac{\sum\limits_i |x_i|}{|\sum\limits_i x_i|} \times \text{ eps}$$

Opérations arithmétiques et erreur – Addition

Théorème de l'Erreur pour les Sommes .- Soit la somme

$$S = x_1 + x_2 + \cdots$$

L'erreur absolue de l'entrée pour cette somme est

$$\Delta^I S = \Delta^I x_1 + \Delta^I x_2 + \cdots$$

et l'erreur de précision de l'entrée est

$$\eta^I\left(S\right) = \frac{x_1}{S}\eta^I\left(x_1\right) + \frac{x_2}{S}\eta^I\left(x_2\right) + \cdots \text{ , avec } \mid \eta^I\left(x_1\right) \mid \leq \text{ eps.}$$

$\Rightarrow$ Erreur maximale de la somme

$$\mid \eta^I\left(S\right) \mid \leq \mid \frac{x_1}{S} \mid\mid \eta^I\left(x_1\right) \mid + \mid \frac{x_2}{S} \mid\mid \eta^I\left(x_2\right) \mid + \cdots \leq \frac{1}{|S|}\left(\mid x_1 \mid + \mid x_2 \mid + \cdots\right) \cdot \text{eps}$$

$$\Rightarrow \ \mid \eta^I\left(S\right) \mid \leq \frac{\sum\limits_i |x_i|}{|\sum\limits_i x_i|} \times \text{ eps}$$

$\Rightarrow$ **Si** $\mid \sum\limits_i x_i \mid$ **est petit par rapport à** $\sum\limits_i \mid x_i \mid \times$ **eps, l'erreur de précision de l'entrée peut devenir importante.**

Opérations arithmétiques et erreur – Multiplication

Théorème de l'Erreur pour les Produits .- Soit le produit

$$P = x_1 \cdot x_2 \cdot \ldots \quad \text{avec } x_i \neq 0 \ \forall i$$

Opérations arithmétiques et erreur – Multiplication

Théorème de l'Erreur pour les Produits .- Soit le produit

$$P = x_1 \cdot x_2 \cdot \ldots \quad \text{avec } x_i \neq 0 \ \forall i$$

L'erreur absolue de l'entrée est

$$\Delta^I P = P \cdot \left(\frac{\Delta^I x_1}{x_1} + \frac{\Delta^I x_2}{x_2} + \cdots \right)$$

Opérations arithmétiques et erreur – Multiplication

Théorème de l'Erreur pour les Produits .- Soit le produit

$$P = x_1 \cdot x_2 \cdot \ldots \quad \text{avec } x_i \neq 0 \ \forall i$$

L'erreur absolue de l'entrée est

$$\Delta^I P = P \cdot \left(\frac{\Delta^I x_1}{x_1} + \frac{\Delta^I x_2}{x_2} + \cdots \right)$$

et l'erreur de précision de l'entrée est

$$\eta^I (P) = \eta^I (x_1) + \eta^I (x_2) + \cdots$$

Opérations arithmétiques et erreur – Multiplication

Théorème de l'Erreur pour les Produits .- Soit le produit

$$P = x_1 \cdot x_2 \cdot \ldots \quad \text{avec } x_i \neq 0 \; \forall i$$

L'erreur absolue de l'entrée est

$$\Delta^I P = P \cdot \left(\frac{\Delta^I x_1}{x_1} + \frac{\Delta^I x_2}{x_2} + \cdots \right)$$

et l'erreur de précision de l'entrée est

$$\eta^I (P) = \eta^I (x_1) + \eta^I (x_2) + \cdots$$

Donc la borne maximale pour l'erreur de précision pour la multiplication est $\eta^I (P) \leq N \cdot \text{eps}$, où N est le nombre de facteurs dans le produit P

Opérations arithmétiques et erreur – Division

THÉORÈME DE L'ERREUR POUR LA DIVISION .- Soit l'opération

$$Q = \frac{a}{b}\ ;\ b \neq 0$$

Opérations arithmétiques et erreur – Division

THÉORÈME DE L'ERREUR POUR LA DIVISION .- Soit l'opération

$$Q = \frac{a}{b}\ ;\ b \neq 0$$

L'erreur absolue de l'entrée est

$$\Delta^I Q = \frac{b\Delta^I a - a\Delta^I b}{b^2}$$

Opérations arithmétiques et erreur – Division

THÉORÈME DE L'ERREUR POUR LA DIVISION .- Soit l'opération

$$Q = \frac{a}{b}\ ;\ b \neq 0$$

L'erreur absolue de l'entrée est

$$\Delta^I Q = \frac{b\Delta^I a - a\Delta^I b}{b^2}$$

et l'erreur de précision à l'entrée est

$$\eta^I\left(Q\right) = \eta^I\left(a\right) - \eta^I\left(b\right)$$

Opérations arithmétiques et erreur – Division

THÉORÈME DE L'ERREUR POUR LA DIVISION .- Soit l'opération

$$Q = \frac{a}{b} \; ; \; b \neq 0$$

L'erreur absolue de l'entrée est

$$\Delta^I Q = \frac{b\Delta^I a - a\Delta^I b}{b^2}$$

et l'erreur de précision à l'entrée est

$$\eta^I (Q) = \eta^I (a) - \eta^I (b)$$

Donc la borne maximale pour l'erreur de précision pour la division est

$$| \eta^I (Q) | \leq 2 \cdot \text{eps}$$