Arithmétique sur ordinateur

Calculer juste ou juste calculer ?

F. Langrognet

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Qui est le responsable ?
3. Que faire ?
4. Références, conclusion

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
   - Des nombres plus dangereux que d'autres
   - Des erreurs parfois importantes
   - Codes équivalents ?
   - Même code => même résultat ?
2. Qui est le responsable ?
3. Que faire ?
4. Références, conclusion

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
   - Des nombres plus dangereux que d'autres
   - Des erreurs parfois importantes
   - Codes équivalents ?
   - Même code => même résultat ?
2. Qui est le responsable ?
3. Que faire ?
4. Références, conclusion

Un calcul simple

vraiment simple...

Calcul de $\sum_{i=1}^{n} x$

Avec $n = 1000$ et

- $x = 0.5$
- $x = 0.25$
- $x = 0.1$
- $x = 0.7$

Un calcul simple

Un peu de C++

```
int main(){
        float x;
        cout<<"Entrez la valeur de x"<<endl;
        cin>>x;
        float res = 0.0;
        for (int i=0;i<1000;i++){
                res+=x;
        }
        cout<<"Somme des x (1000 fois) :"<<setprecision(10)<<res<<endl;
        return 0;
}
```

Un calcul simple

Calculons

$$\sum_{i=1}^{1000} 0.5 = 500$$

$$\sum_{i=1}^{1000} 0.25 = 250$$

$$\sum_{i=1}^{1000} 0.1 = 99.999046$$

$$\sum_{i=1}^{1000} 0.7 = 700.006958$$

Un calcul simple

Calculons

$$\sum_{i=1}^{1000} 0.5 = 500$$

$$\sum_{i=1}^{1000} 0.25 = 250$$

$$\sum_{i=1}^{1000} 0.1 = 99.999046$$

$$\sum_{i=1}^{1000} 0.7 = 700.006958$$

Un calcul simple

Calculons

$$\sum_{i=1}^{1000} 0.5 = 500$$

$$\sum_{i=1}^{1000} 0.25 = 250$$

$$\sum_{i=1}^{1000} 0.1 = 99.999046$$

$$\sum_{i=1}^{1000} 0.7 = 700.006958$$

Un calcul simple

Calculons

$$\sum_{i=1}^{1000} 0.5 = 500$$

$$\sum_{i=1}^{1000} 0.25 = 250$$

$$\sum_{i=1}^{1000} 0.1 = 99.999046$$

$$\sum_{i=1}^{1000} 0.7 = 700.006958$$

Et avec Matlab ?

Ce calcul avec Matlab

```
x = single(0.1);
res = 0.0;
for i=1:1000
        res = res+x;
end;
```

- res = 99.9990
- Pas mieux (même erreur)

Remarque

Tous les calculs (C++ et Matlab) ont été faits en simple précision

un calcul simple

Et avec des double precision ?

On remplace les float par des double et . . .

$$\sum_{i=1}^{1000} 0.1 = 99.999999999985931253$$

On repousse simplement le problème !

un calcul simple

Et avec des double precision ?

On remplace les float par des double et . . .

$$\sum_{i=1}^{1000} 0.1 = 99.99999999985931253$$

On repousse simplement le problème !

Question 1

- Pourquoi les résultats avec $x = 0.5$ et $x = 0.25$ sont-ils justes et ceux avec $x = 0.7$ et $x = 0.1$ faux ?
- 0.7 et 0.1 sont-ils plus dangereux que 0.5 et 0.25 ?

Question 2

Est-ce vraiment grave ?

Exemple concret d'une erreur numérique

- Guerre du Golfe de 1991 : un anti-missile US Patriot dont le programme tournait depuis 100 heures a raté l'interception d'un missile Irakien Scud - 28 morts
- Explication :
  - l'anti missile Patriot incrémentait un compteur toutes les 0.1 secondes
  - 0.1 approché avec erreur 0.0000000953 (codé sur 24 bits)
  - au bout de 100 heures, erreur cumulée 0.34s
  - dans ce laps de temps le Scud parcourt 500 mètres.

PLAN

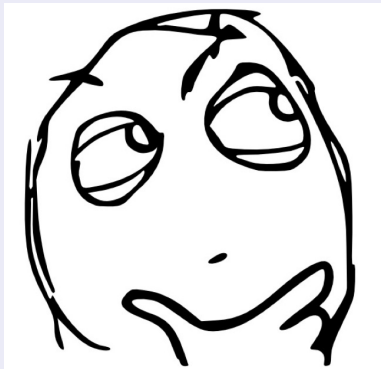
1. Calculer avec un ordinateur, est-ce une bonne idée ?
   - Des nombres plus dangereux que d'autres
   - Des erreurs parfois importantes
   - Codes équivalents ?
   - Même code => même résultat ?
2. Qui est le responsable ?
3. Que faire ?
4. Références, conclusion

Suite de Muller

Soit la suite de Muller définie par :

$$\begin{cases} u_0 = 2 \\ u_1 = -4 \\ u_{n+1} = 111 - \frac{1130}{u_n} + \frac{3000}{u_n . u_{n-1}} \end{cases}$$

- u_n converge vers 6
- Sur n'importe quel système à précision finie on observera une convergence apparente, très rapide, vers 100



Fonction de Rump

Fonction de Rump

$f(a,b) = (333 + \frac{3}{4})b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + \frac{11}{2}b^8 + \frac{a}{2b}$ avec

$a = 77617.0$ et $b = 33096.0$

Résultats

- Simple précision (32 bits) : $\approx -2,34.10^{29}$
- Double précision (64 bits) : $\approx -1,1.10^{21}$
- Avec d'autres parenthésages, sur des architectures différentes
  - Simple précision : $\approx 7,09.10^{29}$ ou encore $\approx +1,17$
  - Double précision : $\approx +1,17$ (souvent)

Quelle est la bonne valeur ?

$$\approx -0.827396059946821368141165095479816291$$

Fonction de Rump

Fonction de Rump

$f(a,b) = (333 + \frac{3}{4})b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + \frac{11}{2}b^8 + \frac{a}{2b}$ avec

$a = 77617.0$ et $b = 33096.0$

Résultats

- Simple précision (32 bits) : $\approx -2,34.10^{29}$
- Double précision (64 bits) : $\approx -1,1.10^{21}$
- Avec d'autres parenthésages, sur des architectures différentes
  - Simple précision : $\approx 7,09.10^{29}$ ou encore $\approx +1,17$
  - Double précision : $\approx +1,17$ (souvent)

Quelle est la bonne valeur ?

$\approx -0.827396059946821368141165095479816291$

Fonction de Rump

Fonction de Rump

$f(a,b) = (333 + \frac{3}{4})b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + \frac{11}{2}b^8 + \frac{a}{2b}$ avec

$a = 77617.0$ et $b = 33096.0$

Résultats

- Simple précision (32 bits) : $\approx -2,34.10^{29}$
- Double précision (64 bits) : $\approx -1,1.10^{21}$
- Avec d'autres parenthésages, sur des architectures différentes
  - Simple précision : $\approx 7,09.10^{29}$ ou encore $\approx +1,17$
  - Double précision : $\approx +1,17$ (souvent)

Quelle est la bonne valeur ?

$\approx -0.827396059946821368141165095479816291$

Fonction de Rump

Fonction de Rump

$f(a,b) = (333 + \frac{3}{4})b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + \frac{11}{2}b^8 + \frac{a}{2b}$ avec

$a = 77617.0$ et $b = 33096.0$

Résultats

- Simple précision (32 bits) : $\approx -2,34.10^{29}$
- Double précision (64 bits) : $\approx -1,1.10^{21}$
- Avec d'autres parenthésages, sur des architectures différentes
  - Simple précision : $\approx 7,09.10^{29}$ ou encore $\approx +1,17$
  - Double précision : $\approx +1,17$ (souvent)

Quelle est la bonne valeur ?

$\approx -0.827396059946821368141165095479816291$

Fonction de Rump

Fonction de Rump

$f(a,b) = (333 + \frac{3}{4})b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + \frac{11}{2}b^8 + \frac{a}{2b}$ avec

$a = 77617.0$ et $b = 33096.0$

Résultats

- Simple précision (32 bits) : $\approx -2,34.10^{29}$
- Double précision (64 bits) : $\approx -1,1.10^{21}$
- Avec d'autres parenthésages, sur des architectures différentes
  - Simple précision : $\approx 7,09.10^{29}$ ou encore $\approx +1,17$
  - Double précision : $\approx +1,17$ (souvent)

Quelle est la bonne valeur ?

$$\approx -0.827396059946821368141165095479816291$$

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
   - Des nombres plus dangereux que d'autres
   - Des erreurs parfois importantes
   - Codes équivalents ?
   - Même code => même résultat ?
2. Qui est le responsable ?
3. Que faire ?
4. Références, conclusion

Codes vraiment équivalents ?

$$f(x) = x^2 + \frac{1}{10} sin(x) \qquad f(x) = x * x + 0.1 sin(x)$$

$$g(n) = \sum_{i=1}^{n} \frac{1}{i} \qquad g(n) = \sum_{i=n}^{1} \frac{1}{i}$$

donnent elles toujours les mêmes résultats ?

2 codes mathématiquement équivalents peuvent mener à des résultats différents

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
   - Des nombres plus dangereux que d'autres
   - Des erreurs parfois importantes
   - Codes équivalents ?
   - Même code => même résultat ?
2. Qui est le responsable ?
3. Que faire ?
4. Références, conclusion

Même code => même résultat ? (1)

Ordre des opérations

```
int main(){
  double x, a, b, c, d;
  x = a + b + c + d;
  return 0;
}
```

Ordre des opérations

- Est-il toujours le même ?
- Est-il indépendant du langage ?
- Est-il indépendant des options du compilateur ?

Le même code peut être exécuté de manières différentes
(mathématiquement équivalentes)

Même code => même résultat ? (2)

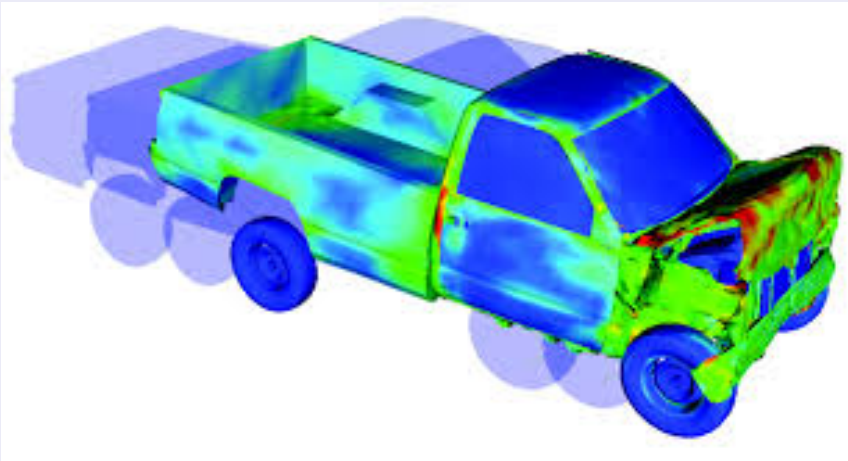
L'ordre des opérations a t-il une importance ?

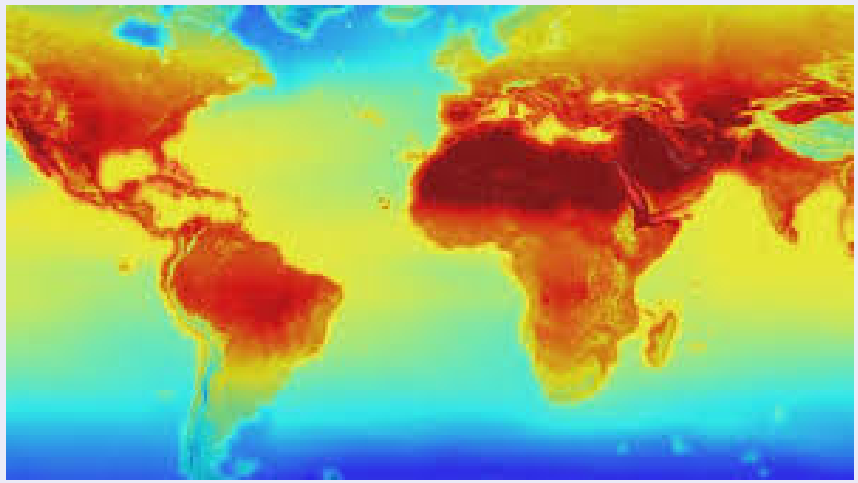
$$(a+b)+c \stackrel{?}{=} a+(b+c)$$

L'ordre des opérations a une importance
Les résultats peuvent être différents

Question 3

Simulation numérique

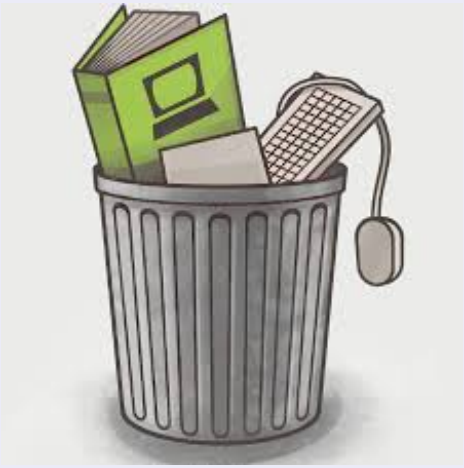




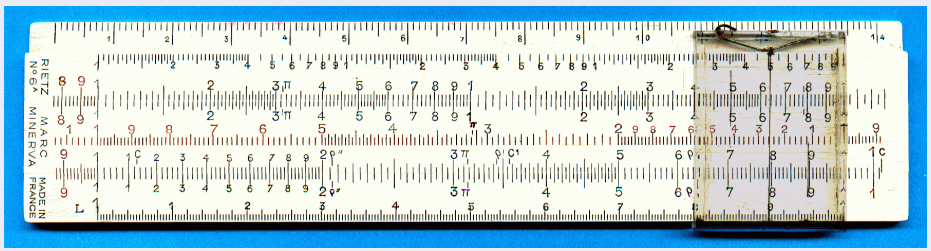
Le climat sera t-il le même si on change de calculateur, de compilateur ?

Question 4

Faut-il jeter son ordinateur à la poubelle ...



... et utiliser la règle à calculs ?



PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Qui est le responsable ?
   - Qui est en charge des calculs ?
   - Que dit la norme ?
   - Les limites de la norme et ses conséquences ?
3. Que faire ?
4. Références, conclusion

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Qui est le responsable ?
   - Qui est en charge des calculs ?
   - Que dit la norme ?
   - Les limites de la norme et ses conséquences ?
3. Que faire ?
4. Références, conclusion

Qui est en charge des calculs ?

Le traitement de la précision des calculs se situe sur
l'ensemble de l'environnement logiciel et matériel

- Processeur
  - réalise des opérations de calcul (Floating Point Unit)
  - gère les exceptions
  - possède ses propres registres (pour stocker des flottants)
- Système d'exploitation
  - transmet les exceptions
  - calcule les opérations (/fonctions) non gérées par le processeur
  - gère le statut du calcul flottant : précision, mode d'arrondi
- Langage de programmation
- Compilateur
- Développeur

Langage de programmation

Les langages de programmation n'ont pas toujours les mêmes règles.
Certains imposent un ordre (C++ sauf avec certaines options...), d'autres non (Fortran)

Exemple

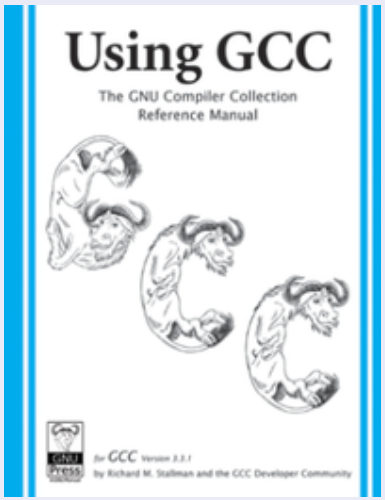
$$x = a + b + c + d$$

- Dans quel ordre seront effectués les calculs ?
- Quel est l'impact sur le résultat ?

Compilateur

Le Compilateur

- possède des centaines d'options
- certaines options permettent de préserver la sémantique du langage
- mais ne sont pas toujours (/rarement ?) activées par défaut
- Marketing : les valeurs par défaut doivent optimiser la vitesse (...et pas la précision)



Développeur

- Est censé maîtrisé tout ce qui précède
- Et ... est tenu pour responsable de tout problème sur le logiciel, y compris la perte de précision

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Qui est le responsable ?
   - Qui est en charge des calculs ?
   - Que dit la norme ?
   - Les limites de la norme et ses conséquences ?
3. Que faire ?
4. Références, conclusion

Avant la norme...

Le chaos !

Norme IEEE 754

La norme en 3 lignes

La version de 1985 définit

- La représentation des réels
  Simple et double précision
- 4 modes d'arrondi
- La notion d'arrondi correct

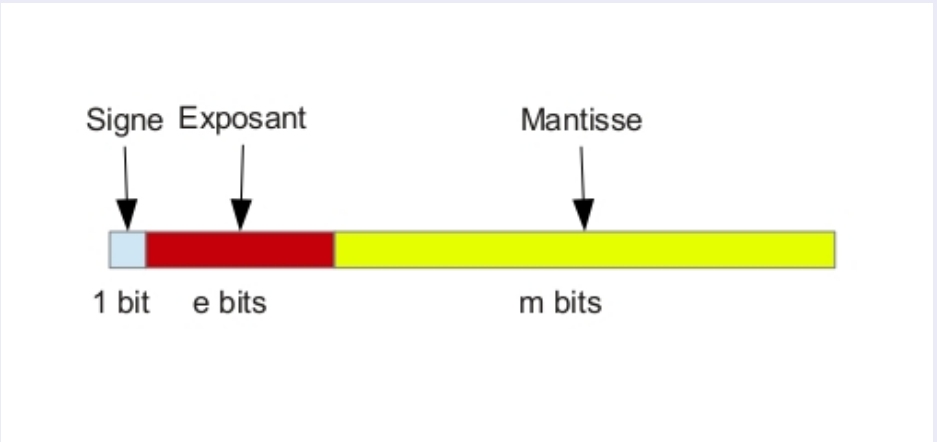
Représentation des réels

Représentation des réels

Mantisse, exposant, signe

Un nombre réel est représenté par un triplet :

- le signe (1 bit)
  Négatif si bit=1, positif sinon
- l'exposant *decalé* (e bits)
  Nombre entier compris entre 0 et $2^e - 1$
- la mantisse (m bits)
  Représente un nombre réel compris entre 0 et 1



Norme IEEE-754 - nombre de bits

Type	Signe	Exposant	Mantisse
float (simple précision)	1	8	23
double (double précision)	1	11	52

Modes d'arrondi

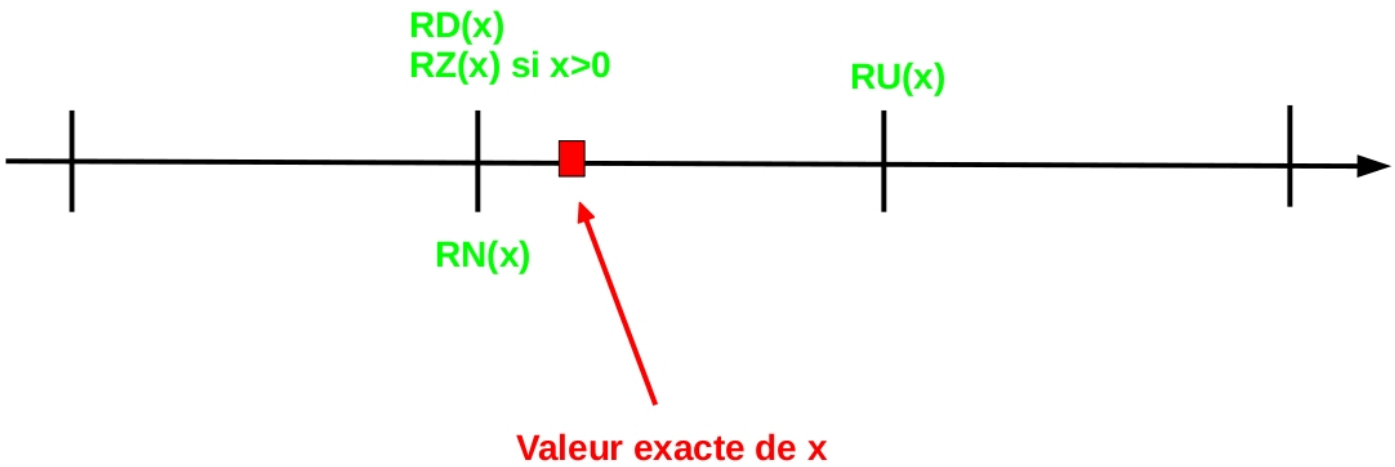
Modes d'arrondi

Arrondis

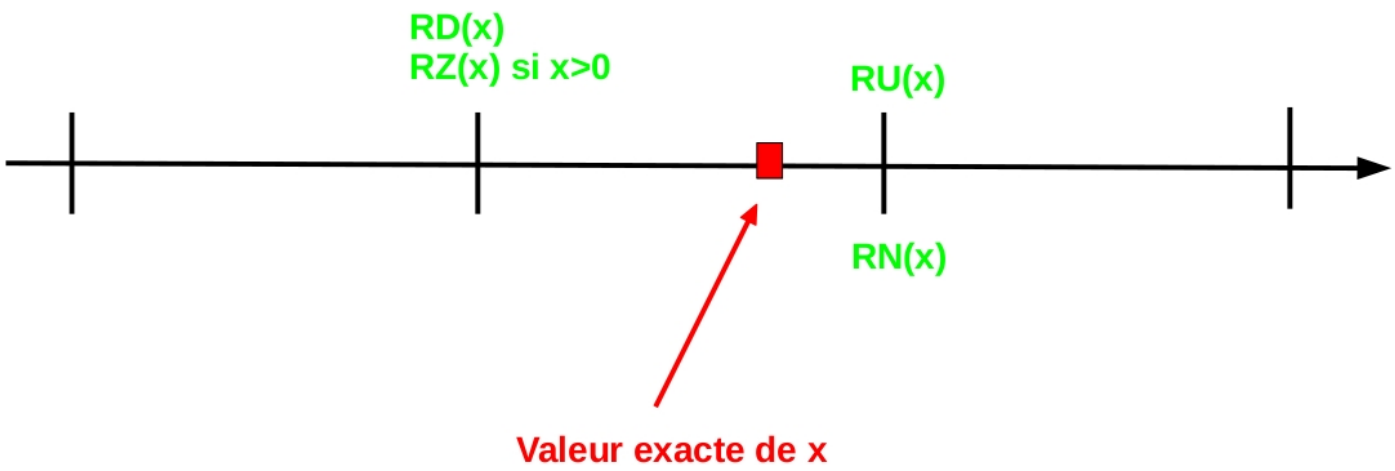
Toutes les opérations (y compris l'affectation) fournissent des valeurs arrondies vers une valeur représentable en machine
Il existe 4 modes d'arrondi dans la nomre IEEE 754 :

- vers moins $+\infty$ (RU)
- vers plus $-\infty$ (RD)
- vers 0 (RZ)
- Au plus proche (RN)

Modes d'arrondi



Modes d'arrondi



Arrondis corrects

Arrondis corrects

Arrondis corrects

Idée : éviter la propagation des erreurs d'arrondis à chaque opération
Soit

- x et y 2 nombres exactement représentables en machine
- $\odot$ une opération
- $\diamond$ le mode d'arrondi

La norme IEEE 754 exige que le resultat d'une operation $x \odot y$ soit égal à $\diamond(x \odot_{exact} y)$
Le résultat doit être le même que si on effectuait le calcul en précision infinie puis on arrondissait ce résultat.

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Qui est le responsable ?
   - Qui est en charge des calculs ?
   - Que dit la norme ?
   - Les limites de la norme et ses conséquences ?
3. Que faire ?
4. Références, conclusion

Absorption

Absorption

Lors d'une addition de 2 nombres ayant des ordres de grandeur très différents, on peut perdre toute l'information du plus petit.

Exemple

$$10^6 + 0.01171875 = 10^6 \text{ !}$$

L'addition respecte bien l'arrondi correct

- 10^6 est le meilleur représentant pour $1000000,01171875$
- Le nombre flottant suivant est $1000000,0625$

Cancellation

Cancellation

Lors d'une soustraction de 2 nombres proches, on introduit des chiffres faux (/inconnus)

Exemple

$9.500000953674316 - 9.5 = 9.53674316\ 40625.10^{-7}$!

La soustraction respecte bien l'arrondi correct

Cancellation et absorption

- Lors d'une cancellation, on introduit arbitrairement des 0 dans la mantisse (car on ne dispose pas de plus de précision sur les opérandes)
- Si les opérandes sont issus d'un calcul précédent ils ont pu subir une perte de précision (en cas d'absorption par exemple)

Cette perte de précision va alors être ré-intégrée et amplifiée.
Dans ce cas, les bits abitrairement introduits (des 0) ne sont pas les bons.

Exemple

$$(10^6 + 0.01171875) - 10^6 = 0$$

Propriétés algébriques dans $\mathbb{F}$

Addition et multiplication

- La commutativité est respectée pour l'addition et la multiplication
  - $a + b = b + a$
  - $a * b = b * a$
- L'associativité n'est pas respectée (en général) ni pour l'addition, ni pour la multiplication
  - $(a + b) + c \stackrel{?}{=} a + (b + c)$
  - $(a * b) * c \stackrel{?}{=} a * (b * c)$
- La distributivité n'est pas respectée (en général) entre la multiplication et l'addition
  - $a(b + c) \stackrel{?}{=} ab + ac$

Arrondis corrects

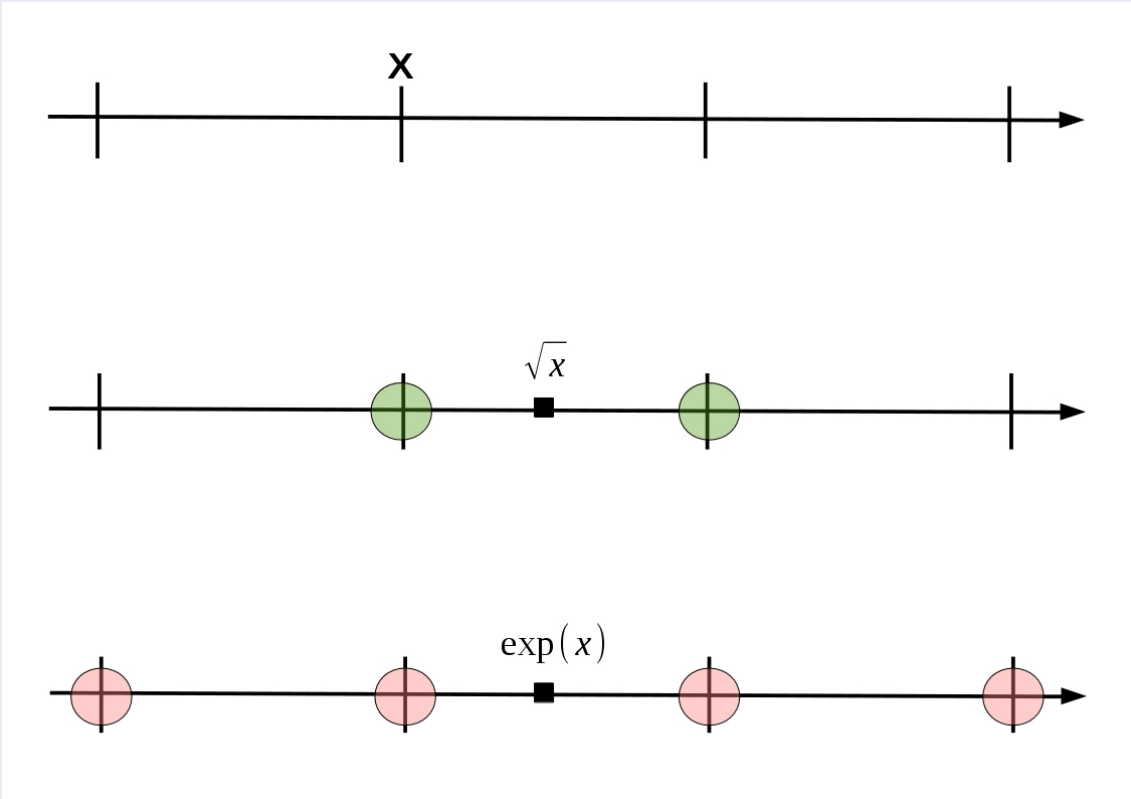
... oui mais pas toujours

- La norme IEEE 754 n'impose l'arrondi correct que pour les opérations : $+, -, *, /, \sqrt{}$
- Pour toutes les autres opérations, la norme conseille l'arrondi correct.

Pour ces fonctions, on ne peut donc jamais être sûr que le résultat d'une opération fournit le meilleur résultat dans $\mathbb{F}$.

L'arrondi correct n'est pas garanti pour une opération mathématique (autre que $+, -, *, /, \sqrt{}$)

Arrondis corrects / non corrects



- Pour $\sqrt{x}$, l'arrondi correct est assuré
  Le résultat est connu (/assuré) (au mode d'arrondi près)
- Pour $exp(x)$, l'arrondi correct n'est pas assuré
  On ne peut pas connaître la valeur retournée

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Qui est le responsable ?
3. Que faire ?
   - Quelques conseils de bon sens
   - Peut-on améliorer ou mesurer la précision ?
4. Références, conclusion

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Qui est le responsable ?
3. Que faire ?
   - Quelques conseils de bon sens
   - Peut-on améliorer ou mesurer la précision ?
4. Références, conclusion

Comparaisons

Comparaisons

La comparaison entre 2 réels doit considérée avec la plus grande prudence !

- Ne jamais tester if $x == y$ ... surtout si x et y sont le résultat d'un calcul précédent
- Attention aux tests d'arrêt if $|x_n - x_{n-1}| < \epsilon$ ... ou même if $\frac{|x_n - x_{n-1}|}{|x_n|} < \epsilon$ ...
- Attention aux algorithmes dépendant de la comparaison entre 2 réels
  Ex : if $(x < y)$ <Traitement1> else <Traitement2>

Savoir ordonner les opérations

... ou comment minimiser les effets de l'absorption

Impact de l'ordre des opérations

Somme des inverses de i

- De 1 à N : $$\sum_{i=1}^{n} \frac{1}{i} \qquad (1)$$
- ou de N à 1 : $$\sum_{i=n}^{1} \frac{1}{i} \qquad (2)$$

Impact de l'ordre des opérations

Somme des inverses de i (de 1 à N)

En simple précision, on obtient :

N	10^5	10^6	10^7	10^8
valeur exacte	12.09015	14.39273	16.69531	18.99790
$1 \rightarrow N$	12.09085	14.35736	15.40368	15.40368
$N \rightarrow 1$	12.09015	14.39265	16.68603	18.80792

- $\sum_{i=1}^{n} \frac{1}{i}$: Pour n grand, on finira par additionner des réels d'ordres de grandeur très différents : absorptions
- $\sum_{i=n}^{1} \frac{1}{i}$: Les ordres de grandeur dans les additions seront semblables

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Qui est le responsable ?
3. Que faire ?
   - Quelques conseils de bon sens
   - Peut-on améliorer ou mesurer la précision ?
4. Références, conclusion

Peut-on améliorer la précision ?

Bibliothèques de calcul utiles

- Flottants codés sur plus de bits
  GMP (GNU Multiple Precision Arithmetic Library)
- Arrondi correct pour toutes les fonctions mathématiques
  CRlibm (Correctly Rounded Mathematical LIBrary)
- Précision arbitraire **et** arrondis corrects pour toutes les fonctions
  MPFR

Peut-on mesurer la précision ?

Arithmétique stochastique

Principe

- En changeant le mode d'arrondi à chaque opération, on propage différemment les erreurs d'arrondi
- En lançant plusieurs fois les calculs, on obtient des résultats différents
- L'étude statistique de ces résultats (représentant un échantillon d'une loi de probabilité) donne des informations sur la précision

Bibliothèque Cadna

La bibliothèque Cadna intégre l'arithmétique stochastique

- Donne le nombre de chiffres significatifs
- Indique les lignes de code où apparaissent des pertes de précision (cancellations, absorptions, branchements instables, ...)

Arithmétique par intervalle

Formule générale

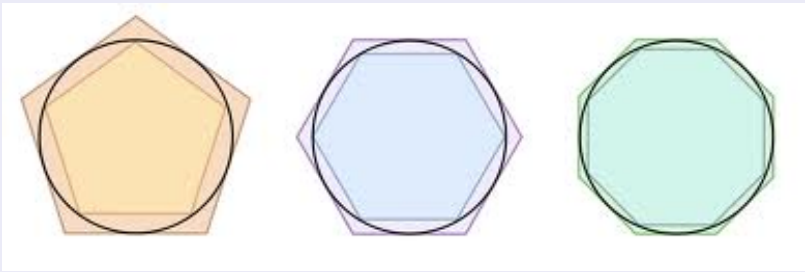
$$X \diamond Y = Hull\{x \diamond y; x \in X, y \in Y\}$$

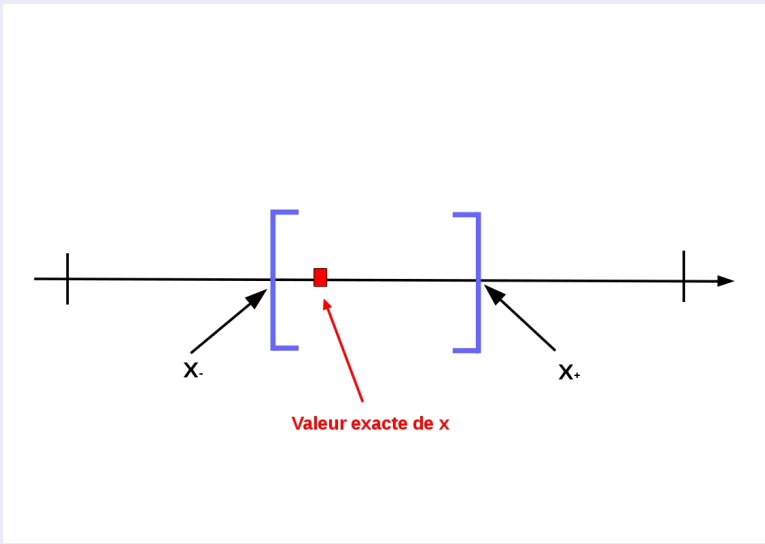
Opérations courantes

$$[\underline{x}, \overline{x}] + [\underline{y}, \overline{y}] = [\underline{x} + \underline{y} \ , \ \overline{x} + \overline{y}]$$

$$[\underline{x}, \overline{x}] - [\underline{y}, \overline{y}] = [\underline{x} - \overline{y} \ , \ \overline{x} - \underline{y}]$$

$\ldots$





MPFI (bibliothèque d'arithmétique par intervalles multi-précision)

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Qui est le responsable ?
3. Que faire ?
4. Références, conclusion

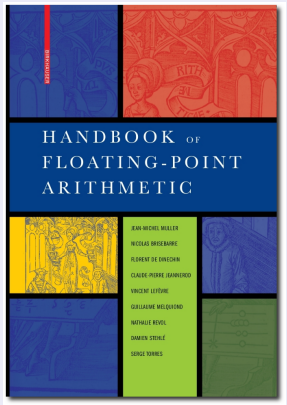
Références

Ma page personnelle

https ://lmb.univ-fcomte.fr/Florent-Langrognet

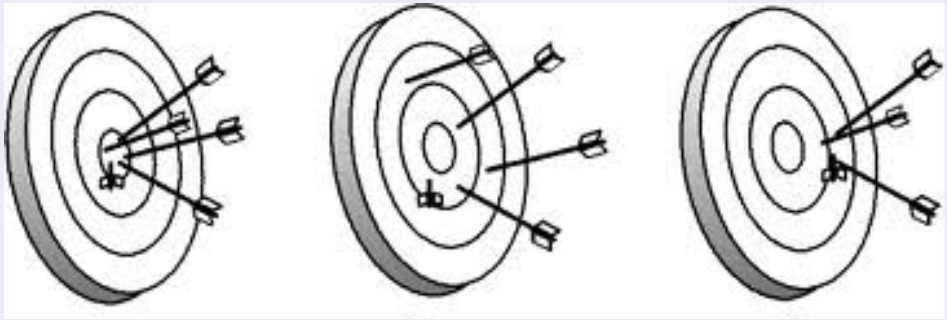
- Un cours pour comprendre la représentation des nombres réels, l'arithmétique flottante, par intervalle et stochastique)
- Un article dans HPC magazine
- Une école "précision et reproductibilité en calcul numérique" (http ://calcul.math.cnrs.fr/spip.php ?article220)

Un livre



Conclusion

Calculer juste ou juste calculer ?



Les risques sont réels...
.. mieux vaut les connaître.