Introduction à l'arithmétique des ordinateurs

Peut-on vraiment calculer avec un ordinateur ?

F. Langrognet

Semaine mathématiques et numérique

Besançon, 29 janvier 2020

PLAN

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
   - Sommes-nous condamnés à n'utiliser que certains nombres ?
   - Des erreurs parfois importantes
   - Fonctions mathématiques et codes équivalents ?
2. Arithmétique flottante
3. Comment améliorer la précision ou la mesurer ?
4. Ouverture et conclusion

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
   - Sommes-nous condamnés à n'utiliser que certains nombres ?
   - Des erreurs parfois importantes
   - Fonctions mathématiques et codes équivalents ?
2. Arithmétique flottante
3. Comment améliorer la précision ou la mesurer ?
4. Ouverture et conclusion

Un calcul simple

vraiment simple...

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

x	C++ (simple précision)	Matlab	C++ (double précision)
0.5	**500**	**500**	**500**
0.25	**250**	**250**	**250**
0.1	99.999046	99.999046	99.9999999999985931253
0.7	700.006958	700.006958	700.0000000000636646

Questions ...

- Sommes-nous condamnés à ne travailler qu'avec des puissances de 2 ?

- Une erreur de 10^{-3} voire 10^{-11}... 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 est approché avec erreur 0.0000000953 (codé sur 24 bits)
  - Au bout de 100 heures, l’erreur cumulée est de 0.34s
  - Dans ce laps de temps le Scud parcourt 500 mètres !

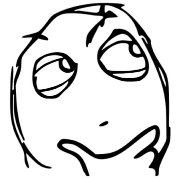
PLAN

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 ($u_n = \frac{3.6^{n+1} - 4.5^{n+1}}{3.6^n - 4.5^n}$)
- Sur n'importe quel système à précision finie on observera une convergence apparente, très rapide, vers 100



PLAN

Fonctions mathématiques équivalentes

Résultats éventuellement différents

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

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

donnent elles toujours les mêmes résultats ?

2 codes mathématiquement équivalents peuvent mener à des résultats différents

Pire encore : le même code peut donner des résultats différents

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 ?

$\approx -0.827396059946821368141165095479816 29$

Pire encore : le même code peut donner des résultats différents

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 ?

$\approx -0.827396059946821368141165095479816 29$

Pire encore : le même code peut donner des résultats différents

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 ?

$\approx -0.827396059946821368141165095479816 29$

Pire encore : le même code peut donner des résultats différents

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 ?

$\approx -0.827396059946821368141165095479816 29$

Pire encore : le même code peut donner des résultats différents

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 ?

$\approx -0.827396059946821368141165095479816 29$

PLAN

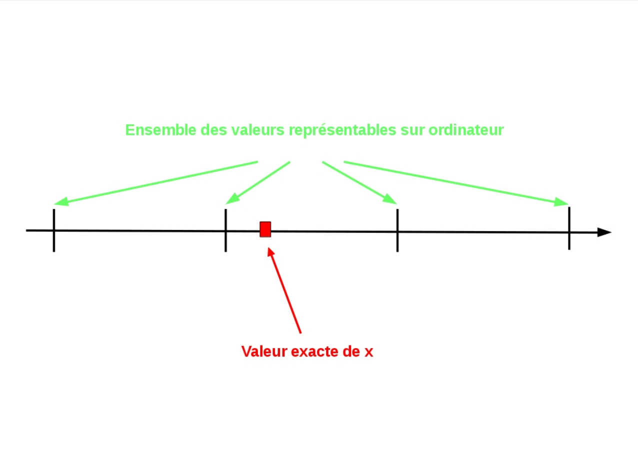
PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Arithmétique flottante
   - Stockage et affectation
   - Arrondir le résultat d'un calcul, calculer avec des nombres arrondis
   - Addition et soustraction
   - Comparaisons
3. Comment améliorer la précision ou la mesurer ?
4. Ouverture et conclusion

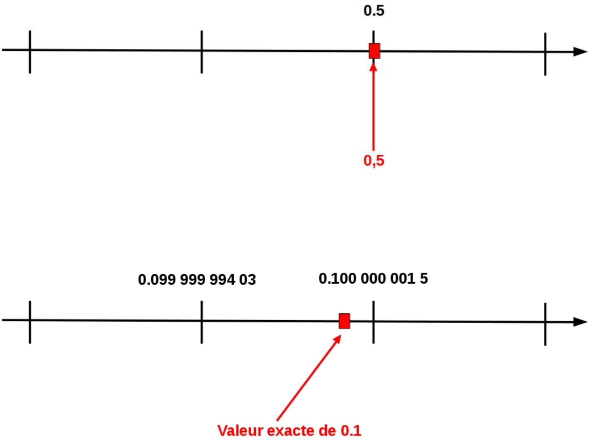
Affectation

Choisir le meilleur représentant

Comment passer de $\mathbb{R}$ à $\mathbb{F}$, ensemble (discret) des nombres flottants ?



Affectation : 1^{re} source d'erreur



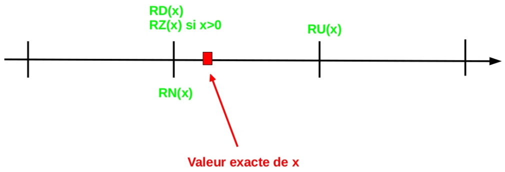
Modes d'arrondis

Arrondis

Toutes les opérations (y compris l'affectation) fournissent des valeurs arrondies vers une valeur représentable en machine

Il existe 4 modes d'arrondi (IEEE 754) :

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



Question ...

La meilleure représentation de x
est-elle toujours la plus proche ?

PLAN

Arrondis corrects

Arrondis corrects

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

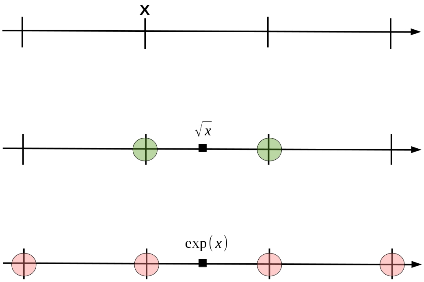
- x et y 2 nombres exactement représentables en machine
- $\odot$ une opération
- $\diamond$ le mode d'arrondi

Arrondi correct : le resultat d'une operation $x \odot y$ doit être 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.

Arrondi correct

Arrondi correct et norme IEEE 754

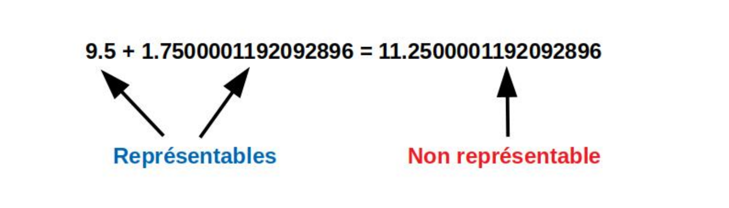
- La norme IEEE 754 n'impose l'arrondi correct que pour les opérations : $+, -, *, /, \sqrt{}$
- Pour toutes les autres opérations, la norme conseille l'arrondi correct. Pour ces fonctions, on ne peut donc jamais être sûr que le résultat d'une opération fournit le meilleur résultat dans $\mathbb{F}$.



PLAN

Addition

Le résultat de l'addition de 2 nombres représentables n'est pas toujours représentable



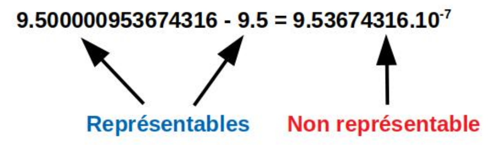
- La valeur calculée est 11.25. C'est la meilleure valeur représentable
- L'addition respecte l'arrondi correct mais il peut y avoir une perte de précision

Absorption : lorsque la perte de précision est de l'ordre d'un des opérandes

$$10^6 + 0.01171875 = 10^6$$

Soustraction

Le résultat de la soustraction de 2 nombres représentables n'est pas toujours représentable



- La valeur calculée est 9.53674316**40625**$.10^{-7}$. C'est la meilleure valeur représentable
- La soustraction respecte l'arrondi correct mais il peut y avoir une perte de précision

Cancellation : lorsque l'on soustrait 2 nombres proches

$$9.500000953674316 - 9.5 = 9.53674316\mathbf{40625}.10^{-7}$$

Cancellation et absorption

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

... alors que l'obtient $(10^6 - 10^6) + 0.01171875 = 0.01171875$

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

Impact de l'ordre des opérations sur le résultat

... 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

Comparaisons

Danger

La comparaison entre 2 réels doit considérée avec la plus grande prudence surtout s'ils sont le résultat d'un calcul précédent

Illustration : Equation du 2^e degré

$$0.3x^2 + 2.1x + 3.675 = 0$$

- Avec le mode d'arrondi par défaut (au plus près), on obtient $\Delta = -9.53674.10^{-7}$
- Le test if $(\Delta == 0)$ conduit à 2 racines complexes
- Alors que 3.5 est racine double

Quelles solutions face à ces problèmes numériques ?

A chaque problème, il y a une solution...ou pas

« Sergent ! interroge Camember, et la terre du trou ? — Que vous êtes donc plus herméfitiquement bouché qu'une bouteille de limonade, sapeur ! Creusez un autre trou ! — C'est vrai ! » approuve Camember.

Le sergent Bitur au sapeur Camember :
« M'ferez quatre jours pour n'avoir pas creusé le deuxième trou assez grand pour pouvoir y mettre sa terre avec celle du premier trou. »

PLAN

PLAN

Précision arbitraire et/ou arrondi correct généalisé

Précision arbitraire

- Utiliser des flottants codés sur plus de bits
- Tous les problèmes de l'arithmétique flottante sont toujours présents.
  On *"repousse"* les problèmes.
- Exemple : GMP (GNU Multiple Precision Arithmetic Library) - http ://gmplib.org

Arrondi correct généralisé

- Etendre l'arrondi correct pour toutes les fonctions
- Calculs parfois très coûteux et complexes
- Exemple : CRlibm (Correctly Rounded Mathematical LIBrary) - http ://lipforge.ens-lyon.fr/www/crlibm/

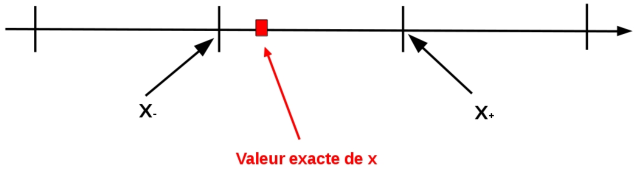
Précision arbitraire **et** arrondi correct généralisé

- Utiliser des flottants codés sur plus de bits
- Etendre l'arrondi correct pour toutes les fonctions
- Exemple : MPFR - http ://www.mpfr.org/

PLAN

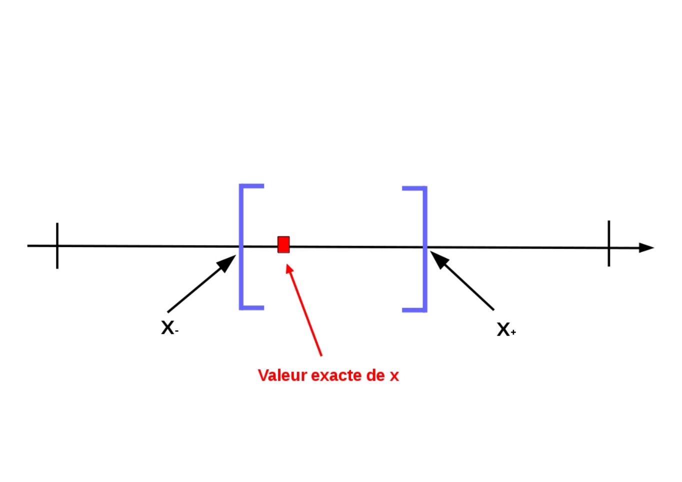
Principe

Comment choisir la meilleure représentation de x ?



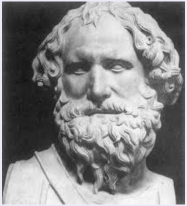
Principe

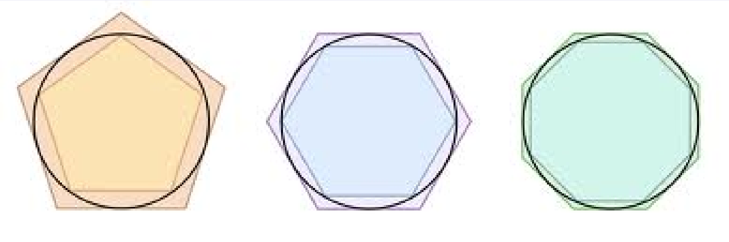
Et si on choisissait l'intervalle ?



Une idée pas si nouvelle...

Archimède utilisait l'arithmétique par intervalles





Formalisation

Formule générale

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

Opérations courantes

$$[\underline{x}, \overline{x}] + [\underline{y}, \overline{y}] = [\underline{x} + \underline{y}\ ,\ \overline{x} + \overline{y}]$$
$$[\underline{x}, \overline{x}] - [\underline{y}, \overline{y}] = [\underline{x} - \overline{y}\ ,\ \overline{x} - \underline{y}]$$
$$[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}] = [min(\underline{x} \times \underline{y}\ ,\ \underline{x} \times \overline{y}\ ,\ \overline{x} \times \underline{y}\ ,\ \overline{x} \times \overline{y})\ ,\ max(id)]$$
$$\begin{aligned} [\underline{x}, \overline{x}]^2 &= [min(\underline{x}^2, \overline{x}^2)\ ,\ max(\underline{x}^2, \overline{x}^2)]\ si\ 0 \notin [\underline{x}, \overline{x}] \\ &= [0, max(\underline{x}^2, \overline{x}^2)]\ sinon \end{aligned}$$

MPFI

Bibliothèque d'arithmétique par intervalles multi-précision

Avantages et inconvénients

Avantage

- **Le vrai résultat est contenu dans l'intervalle résultat calculé**

Inconvénients

- Sur-estimation : l'intervalle résultat peut être (très) grand
  Ex : Fonction de Rump : $[-3.89595.10^{22}, 3.65983.10^{22}]$ contient le résultat exact (-0.8273...).
- Efficacité
  La nécessité de changer souvent le mode d'arrondi a un impact important sur l'efficacité
- La soustraction n'est pas l'opération inverse de l'addition
  Ex : $X - X \neq [0]$
- La division n'est pas l'opération inverse de la multiplication
- $[\underline{x}, \overline{x}]^2 \neq [\underline{x}, \overline{x}] \times [\underline{x}, \overline{x}]$
- Autres difficultés : comparaison d'intervalles, ...

PLAN

1. Calculer avec un ordinateur, est-ce une bonne idée ?
2. Arithmétique flottante
3. Comment améliorer la précision ou la mesurer ?
   - Précision arbitraire et/ou arrondi correct généralisé
   - Arithmétique par intervalles
   - Arithmétique stochastique
4. Ouverture et conclusion

Quel est l'impact du choix du mode d'arrondi sur le résultat ?

Quel est l'impact du choix du mode d'arrondi sur le résultat ?

Rappel

Choisir un mode d'arrondi, c'est choisir un représentant $X \in \mathbb{F}$ de $x \in \mathbb{R}$.
Il existe 4 modes d'arrondi.

Exemple

$$\sum_{i=1}^{n} 0.1$$

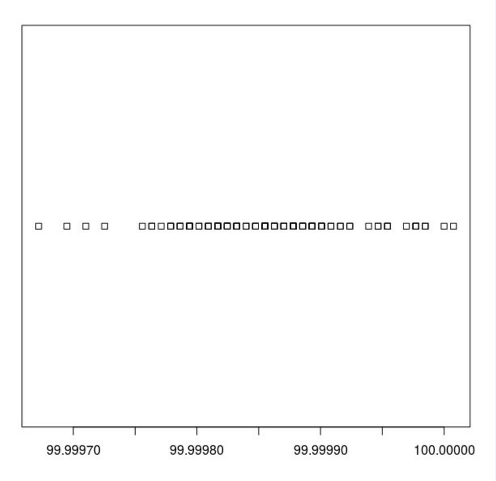
Mode d'arrondi	Résultat	Erreur
Au plus près (RN)	99.99904633	10^{-3}
vers $+\infty$ (RU)	100.0030442	3.10^{-3}

Et avec un mode d'arrondi aléatoire ?

Résultat pour 100 essais

En changeant le mode d'arrondi (aléatoirement) à chaque calcul

- Erreur : de 3.10^{-4} à 0 !
- Les résultats sont globalement meilleurs
- Il arrive que l'on trouve le bon résultat



Arithmétique stochastique

En une page...

- Objectif : n'est pas d'améliorer la précision du résultat mais de l'estimer.
- Comment ? : Obtenir un échantillon de résultats (en propageant différemment les erreurs d'arrondi) et utiliser les statistiques.
- Résultats statistiques :
  - Sous quelques hypothèses, on montre que les résultats calculés suivent une loi gaussienne dont l'espérance est le résultat mathématique exact
  - Par un test de Student, on montre que l'on a une estimation du nombre de chiffres significatifs du résultat avec 3 exécutions seulement (intervalle de confiance à 95%)

La bibliothèque Cadna

Cadna

- Estimation de la qualité numérique d'un résultat
- Contrôle des instructions de branchement
- Détection des cancellations et des absorptions

http ://www.lip6.fr/cadna

En pratique

- Tous les réels (float et double) sont remplacés par des types de Cadna. Chaque réel est représenté par 3 valeurs différentes.
- Pour chaque calcul, on change aléatoirement le mode d'arrondi.
- La précision est donnée par le nombre de chiffres communs entre les 3 valeurs du résultat
  Exemple : (1.234 ; 1.236 ; 1.21) => Le résultat exact est 1.2 (à 10^{-1} près à 95% de confiance)

1^{er} exemple : $\sum_{i=1}^{n} 0.1$

```
CADNA_C 1.1.9 software ---
University P. et M. Curie --- LIP6
Self-validation detection: ON
Mathematical instabilities detection: ON
Branching instabilities detection: ON
Intrinsic instabilities detection: ON
Cancellation instabilities detection: ON
-----------------------------------------
Entrez la valeur de x
0.1
Somme des x (1000 fois) : 0.100000E+003
-----------------------------------------
...
No instability detected
-----------------------------------------
```

On peut donc affirmer que le résultat est 100 avec une précision d'au moins 10^{-3}

2^e exemple : $0.3x^2 + 2.1x + 3.675 = 0$

```
d = @.0
Discriminant is zero.
The double solution is 0.349999E+01
---------------------------------------
...
There are 1 numerical instabilities
0 UNSTABLE DIVISION(S)
0 UNSTABLE POWER FUNCTION(S)
0 UNSTABLE MULTIPLICATION(S)
0 UNSTABLE BRANCHING(S)
0 UNSTABLE MATHEMATICAL FUNCTION(S)
0 UNSTABLE INTRINSIC FUNCTION(S)
1 UNSTABLE CANCELLATION(S)
```

Avec Cadna, l'imprécision sur Δ est telle que l'on ne peut pas le distinguer de la valeur 0.
Le test *if* Δ*==0* est vrai et on choisit la bonne branche.
Le résultat est correct.

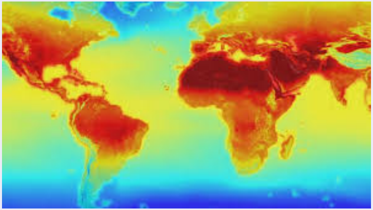
PLAN

Impact de l'environnement informatique

L'ensemble de l'environnement logiciel et matériel a un rôle sur la précision de calcul

- Processeur
- Système d'exploitation
- Langage de programmation
- Compilateur
- Développeur

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



Science reproductible

Reproductibilité

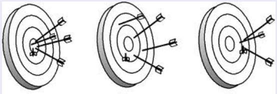
Donner les moyens de reproduire les résultats de la recherche

- La reproductibilité est l'un des fondements de la recherche.
- La reproductibilité concerne toutes les disciplines (biologie, physique, ...).

Reproductibilité numérique

indispensable mais...complexe

- L'ensemble de l'environnement logiciel et matériel peut avoir un impact sur les résultats et donc sur leur reproductibilité.
- Les questions de reproductibilité et de précision sont liées (arithmétique flottante) mais sont différentes.
  Un résultat peut être parfaitement reproductible mais totalement faux.

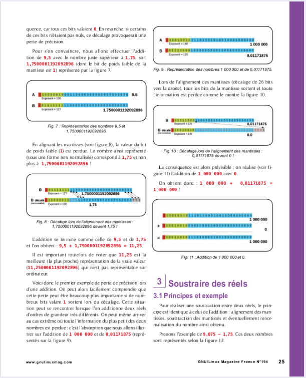


Peut-on vraiment calculer avec un ordinateur ?

Quelques liens

- https ://lmb.univ-fcomte.fr/Florent-Langrognet
  - Séminaire IREM de Franche Comté (Novembre 2015) sur ce thème
  - Article dans HPC Magazine
- Série d'articles dans Linux Magazine 2016

  Num 193, 194, 197
- Ecoles thématiques CNRS PRECIS (Précision, REproductibilité en Calcul et Informatique Scientifique)
  - 2013 : https ://calcul.math.cnrs.fr/2013-03-ecole-precis.html
  - 2017 : https ://calcul.math.cnrs.fr/2017-05-ecole-precis.html



quence, car tous ces bits valaient 0. En revanche, si certains de ces bits n'étaient pas nuls, ce décalage provoquerait une perte de précision.

Pour s'en convaincre, nous allons effectuer l'addition de 9,5 avec le nombre juste supérieur à 1,75, soit 1,7500001192092896 (dont le bit de poids faible de la mantisse est 1) représenté par la figure 7.

Fig. 7 : Représentation des nombres 9,5 et 1,7500001192092896.

En alignant les mantisses (voir figure 8), la valeur du bit de poids faible (1) est perdue. Le nombre ainsi représenté (sous une forme non normalisée) correspond à 1,75 et non plus à 1,7500001192092896 !

Fig. 8 : Décalage lors de l'alignement des mantisses : 1,7500001192092896 devient 1,75 !

L'addition se termine comme celle de 9,5 et de 1,75 et l'on obtient : 9,5 + 1,7500001192092896 = 11,25.

Il est important toutefois de noter que 11,25 est la meilleure (la plus proche) représentation de la vraie valeur (11,2500001192092896) qui n'est pas représentable sur ordinateur.

Voici donc le premier exemple de perte de précision lors d'une addition. On peut alors facilement comprendre que cette perte peut être beaucoup plus importante si de nombreux bits valant 1 sortent lors du décalage. Cette situation peut se rencontrer lorsque l'on additionne deux réels d'ordres de grandeur très différents. On peut même arriver au cas extrême où toute l'information du plus petit des deux nombres est perdue : c'est l'absorption que nous allons illustrer sur l'addition de 1 000 000 et de 0,01171875 (représentés sur la figure 9).

Fig. 9 : Représentation des nombres 1 000 000 et de 0,01171875.

Lors de l'alignement des mantisses (décalage de 26 bits vers la droite), tous les bits de la mantisse sortent et toute l'information est perdue comme le montre la figure 10.

Fig. 10 : Décalage lors de l'alignement des mantisses : 0,01171875 devient 0 !

La conséquence est alors prévisible : on réalise (voir figure 11) l'addition de 1 000 000 avec 0.

On obtient donc : 1 000 000 + 0,01171875 = 1 000 000 !

Fig. 11 : Addition de 1 000 000 et 0.

3 Soustraire des réels

3.1 Principes et exemple

Pour réaliser une soustraction entre deux réels, le principe est identique à celui de l'addition : alignement des mantisses, soustraction des mantisses et éventuellement renormalisation du nombre ainsi obtenu.

Prenons l'exemple de 9,875 − 1,75. Ces deux nombres sont représentés selon la figure 12.

www.gnulinuxmag.com GNU/Linux Magazine France N°194 25

florent.langrognet@free.fr