Bibliographie

- *Qualité des calculs sur ordinateurs, vers des arithmétiques plus fiables* ? Marc Daumas, Jean-Michel Muller, edt MASSON mars 1997

Plan

- Cas d'école de calculs faux
- Des exemples célèbres
- Norme IEEE 754 de 1985
- Analyse des erreurs

Calcul faux !

- 1/3 ≈ 0,33333… suite de 3 infinie
- Faire un arrondi ⇒ perte de l'exactitude
- Exemple : 2/3 arrondi à 0,67€

  2/3 + 2/3 arrondi à 0,67 + 0,67 = 1,34€

  or 2/3 + 2/3 = 4/3 ≈ 1,33333… arrondi à 1,34€

  Il y a donc 1 centime de plus !
- Conclusion :

  4/3 différent <u>numériquement</u> de 2/3 + 2/3

  le cumul des arrondis ⇒ erreur

Encore plus faux !

$r = 9\,x^4 - y^4 + 2\,y^2$ avec x=10864 et y=18817

<u>Résultats :</u> machines norme VF IEEE 754 (1985)

- simple précision (32 bits) : $r = 708\ 158\ 976$

  $9\,x^4 = y^4 = 1{,}253722\underline{9}\ 10^{17}$ et $r = 2\,y^2 = 70815897\underline{6}$

- double précision (64 bits) : $r = 2$ (sous Excel)

  $9\,x^4 = 1{,}25372283822342\underline{1}\ 10^{17}$ approché

  $y^4 = 1{,}25372284530501\underline{1}\ 10^{17}$ approché

  $9\,x^4 - y^4 = -70815897\underline{6}$ et $2\,y^2 = 70815897\underline{8}$ exact

- quadruple précision (128 bits) : $r = 1$ EXACT

  $9\,x^4 = 1253722838223421\underline{44}$ exact

  $y^4 = 1253722845305011\underline{21}$ exact

  $9\,x^4 - y^4 = -70815897\underline{7}$ et $2\,y^2 = 70815897\underline{8}$

Bits et octets

- 1 bit = 1 état du processeur valeurs 1 ou 0 (binaire)
- 1 octet = 8 bits soit 2^8=256 valeurs différentes 00100110
- Simple précision = 32 bits = 4 octets
  Soit 2^{32}=4 294 967 296 valeurs différentes

- Usage traditionnel : erroné selon SI !
  - 1 ko (kilo-octet) = 2^{10} octets = 1024 octets
  - 1 Mo (méga-octet) = 1024 ko = 1024^2 octets
  - 1990 → 1 Go (giga-octet) = 1024 Mo = 1024^3 octets
  - 2000 → 1 To (téra-octet) = 1024 Go = 1024^4 octets (4 = **tétra**)
  - 2010 → 1 Po (péta-octet) = 1024 To = 1024^5 octets (5 = **penta**)

Bits et octets (suite)

- Normalisation SI (1998) :

  1 kibioctet (Kio) = 2^{10} octets = 1024 octets

  (mébi=2^{20}, gibi, tébi, pébi, exbi, zébi et yobi=2^{80})

  1 kilooctet (ko) = 10^{3} octets = 1000 octets

  (méga, giga, téra, péta, exa, zetta et yotta=10^{24})

- Autres usages :

  Mots sur plusieurs octets (stockage)

  Bytes (anglais noté B) ensemble de bits (noté b) adjacents,

  1 byte = 1 octet en général

- Flops : opérations en virgule flottante / seconde

  Machines : 4 gigaflops (PC) $\Rightarrow$ pétaflops

Quelques grands nombres

- Théorie(s) des cordes (cosmologie)
  - 11 dimensions : 3 droites + 1 temps + 7 boucles
  - 10^{500} univers différents pour la genèse de l'espace-temps !

- Entropie S : mesure du désordre ($dS \geq 0$)
  - $S = k \, Ln \, N$, k cste de Boltzmann et N états
  - Big Bang, inflation (t=0) : $S \sim 10^{10}$ -10^{15}
  - Rayonnement fossile (t=380000 ans) : $S \sim 10^{88}$
  - Aujourd'hui ($t=14 \times 10^9$ ans) : $S \sim 10^{100}$
  - "mort" thermique ($t=10^{100}$ ans) : $S_{max} \sim 10^{123}$

cosmos

↓

chaos

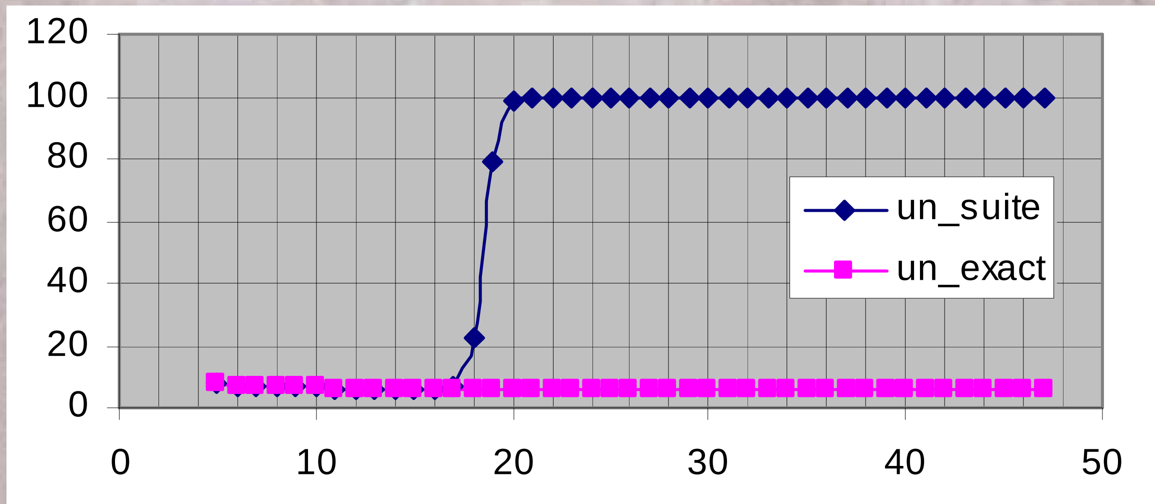
Toujours faux !

Suite $(u_n) = \{u_0, u_1, u_2, \ldots, u_n\}$

$u_0=2$; $u_1=-4$; $u_{n+1}=111 - 1130/u_n + 3000/(u_n \times u_{n-1})$ s1

u_n converge vers 6 : $u_n=(4x5^{n+1}-3x6^{n+1})/(4x5^n-3x6^n)$

Mais numériquement converge vers 100 sur toute les machines, pour toute précision !



calcul Excel (64 bits)

Diapositive 8

s1 l'équation U^3 - 111 U^2 + 1130 U -3000 = 0 a trois solutions U=5, U=6 et U=100

scetash; 15/03/2010

Les erreurs

Phénomène physique

lois
données

Phénomène mathématique

équations
discrétisation

Phénomène informatique

machines
opérations
arrondis

+ l'Homme : errare humanum est
conception, étourderie

Exemple 1

- Bourse de Vancouver (années 80)

En 1982, nouvel indice avec valeur nominale de 1000

A chaque transaction : recalcul de l'indice + troncature au 3ème chiffre décimal

Au 22ème mois : indice calculé = 524,881 au lieu 1098,811 !

⇒ l'erreur de troncature diminuait à chaque fois l'indice (toujours dans le même sens)

Exemple 2

- Guerre du Golfe (1991) : missile anti-missile "Patriot" rate sa cible et fait 28 morts. Cause : les erreurs d'arrondis sur le temps de parcours

Nombre sur 24 bits , horloge interne système 1/10 sec

Or 1/10 = 0,1 = 0,<u>000110011001100110011001</u>1… (binaire) arrondi à 24 chiffres donc erreur à chaque 1/10 sec

Au moment de l'attaque, la batterie du Patriot était allumait depuis 100 heures d'où une erreur cumulée de 0,34 sec. Le missile Scud Irakien avait parcourt 500m !

<u>Solution</u> : redémarrage régulier le système de guidage

Exemple 3

- Explosion d'Ariane 5 : 4 juin 1996

Premier lancement, explosion 34 sec après l'allumage, coût 500 M$

Erreur de programmation dans le système inertiel de référence. La vitesse horizontale de la fusée par rapport à la plate-forme de tir codée sur 64 bits était convertie en un entier de 16 bits alors qu'il était > 32768 (overflow)

Exemples …

- Prévision du temps : Météo

Phénomènes turbulents, chaotiques

Chaos numérique

- Bug de l'an 2000 : date sur 2 chiffres

Mise à jour des ordinateurs, des logiciels, …

Coût important pour les états, les banques, entreprises, … 100 à 200 MM$ en Europe !

Prochain bug en 2038 système Unix 32 bits, décompte les secondes depuis 1er jan 1970

Pbm de normalisation des dates

La norme IEEE 754

- Représentation en "virgule flottante"

$$x \approx s_x \times m_x \times b^{e_x}$$

avec s_x *signe* ±1

$0 \leq m_x < b$ *mantisse* en base b sur n chiffres

e_x *exposant* tq $e_{min} \leq e_x < e_{max}$

- ANSI/IEEE Standard 754-1985

1. Hiérarchie de types (codage homogène)
2. Clôture des opérations (±Inf, Zéro±, NaN, …)
3. Reproductibilité des calculs et arrondi exact :

Indépendance des caractéristiques de la machine

4 modes d'arrondis

La norme IEEE 754 (2)

- Types de formats :
  - simple précision (32 bits) : 1 bit de signe, 8 bits d'exposant (-126 à 127) et 23 bits de mantisse (10^7)

Cf page suivante

Ex) -118,625 (dec) = 1100 0010 1110 1101 0100 0000 0000 0000 (fl)

  - Simple précision étendue (>42 bits) : peu utilisé
  - double précision (64 bits) : 1 bit de signe, 11 bits d'exposant (-1022 à 1023) et 52 bits de mantisse (10^{15})
  - double précision étendue (>78 bits)

- Révision majeure du standard depuis juin 2008

-118,625

Exemple) simple précision = 32 bits = 8 octets

-118,625 (dec) = 1100 0010 1110 1101 0100 0000 0000 0000 (float)

- Le signe : nombre négatif donc s=1 (1 bit)
- 118,625 en binaire s'écrit 1110110,101
- Décalons la virgule vers la gauche :

  $$1110110{,}101 = 1{,}110110101 \times 2^6$$

- À gauche : bit implicite avant la virgule

  à droite 14 bits de 0 pour compléter

  m=110 1101 0100 0000 0000 0000 (23 bits)
- Exposant 6 + décalage 127 = $2^{8-1}-1$

  $6 + 127 = 133 \Rightarrow$ en binaire 1000 0101 (8 bits)

caractères ASCII

- ASCII = American Standard Code for Information Interchange (1960)

- 7 bits soit 128 caractères dont 95 imprimables :

  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

- Evolutions :
  - ISO-8859-1 : codage sur 8 bits (1 octet) extension ASCII
  - Unicode/UCS : (déc 2007) codages UTF-8, UFT-16, UTF-32

    US-ASCII compatible UTF-8