Codage des nombres non entiers

M. Combacau
combacau@laas.fr

UNIVERSITÉ TOULOUSE III PAUL SABATIER

NORTHEASTERN UNIVERSITY

11 novembre 2024

Objectif

Savoir coder un nombre non entier en base 2
Connaître la précision d'un code
Comprendre les arrondis et les erreurs de codage

Principe du codage en virgule flottante (1)

- Mot code constitué de trois champs : signe, exposant, mantisse

b_{n-1}	$b_{n-2} \quad \ldots \quad b_p$	$b_{p-1} \quad \ldots \quad b_0$
signe	exposant	mantisse

- Valeur du code :$(-1)^{b_{n-1}} \times (1 \text{ ou } 0), \text{mantisse} \times 2^{exposant}$
- Le 1 devant la mantisse n'est pas représenté dans le code
- Signe : $b_{n-1} = 0 \leftrightarrow$ positif, $b_{n-1} = 1 \leftrightarrow$: négatif
- Exposant : peut être négatif (ca2 ou biaisé)
- Mantisse : virgule fixe avec partie entière = 1 ou 0
  1 valeur normalisée, 0 hors norme (cas particuliers)

Principe du codage en virgule flottante (2)

- Exemple : -12.06.
- Le bit de signe du code est à 1.
- Valeur de l'exposant déduite du nombre de divisions nécessaires pour ramener le nombre à coder sous la forme 1,mantisse.

$$12,06 = 1.5075 \times 2^3$$

- le nombre -12.06 sera donc codé par

1	code de 3	code de 0.5075

La suite dépend des caractéristiques du code utilisé

Précision du code en virgule flottante

Rappel :

b_{n-1}	$b_{n-2} \quad \dots \quad b_p$	$b_{p-1} \quad \dots \quad b_0$
signe	exposant = exp+biais	mantisse

- Précision absolue : Pa=$2^{-p} \times 2^{exp}$ dépend du nombre
- Précision relative :

$$Pr = \frac{2^{-p} \times 2^{exp}}{1, mantisse \times 2^{exp}} = \frac{2^{-p}}{1, mantisse}$$

La mantisse varie entre 1 et 2 $\Rightarrow Pr \in [2^{-p-1}, 2^{-p}]$
La précision relative varie peu sur l'intervalle de codage

Caractéristiques du codage IEEE p754 (1)

- Bit de signe : sans commentaire
- Exposant sur e bits biaisé de $2^{e-1}-1$
- Mantisse est sur m bits, sous la forme (1 ou 0), *mantisse*
- Conventions pour limites et indéterminations
  1. *Exposant* $= 0$, *mantisse* $\neq 0$, nombre dénormalisé
     valeur$=(-1)^{b_{n-1}} \times 0,$ *mantisse* $\times 2^{-(2^{e-1}-1)}$
  2. *Exposant* $= 0$ et *mantisse* $= 0$, code de 0
  3. *Exposant* $= [1..1]$ et *mantisse* $= 0$, code de ∞
  4. *Exposant* $= [1..1]$ et *mantisse* $\neq 0$ code de **NaN**
  * Deux codes de 0 et de ∞ en fonction du bit de signe
  * NaN (Not a Number) : résultat de $0/0$, $\infty \times 0$ ou encore $\sqrt{-1}$
  * *exposant* $= 0 \Rightarrow$ nombre voisin de 0, limite du code
- Codage utilisé par le compilateur libre gcc

Caractéristiques du codage IEEE p754 (2)

Trois formats existent : simple, double et long double

- Simple précision (32 bits), biais exposant : 127

b_{31}	$b_{30} \ldots b_{23}$	$b_{22} \ldots b_0$
signe	exposant	mantisse

- Précision relative : $Pr \approx 2^{-23} \approx 1,2 \times 10^{-7}$
- Plus petit nombre codé (dénormalisé) :
  $exposant = 0$, $mantisse = 2^{-23}$,
  valeur $= 2^{-23} \times 2^{-127} \approx 7.10^{-46}$
- Plus grand nombre codé :
  $mantisse = 1,1...1 = 2 - 2^{-23} \approx 2$, $exposant = 128$
  valeur $\approx 2 \times 2^{128} = 2^{129} \approx 6,8 \times 10^{38}$

Caractéristiques du codage IEEE p754 (3)

- Double précision (64 bits), biais exposant : 1023

b_{63}	b_{62} ... b_{52}	b_{51} ... b_0
signe	exposant	mantisse

- Précision relative : $Pr \approx 2^{-52} \approx 2,2 \times 10^{-16}$
- Plus petit nombre codé (dénormalisé) :
  $exposant = 0$, $mantisse = 2^{-53}$,
  valeur $= 2^{-53} \times 2^{-1023} \approx 7,7.10^{-326}$
- Plus grand nombre codé :
  $mantisse = 1,1...1 = 2 - 2^{-23} \approx 2$, $exposant = 1024$
  valeur $\approx 2 \times 2^{1024} = 2^{1025} \approx 3,6 \times 10^{308}$

Caractéristiques du codage IEEE p754 (4)

- Double précision étendue (80 bits), biais exposant : 16383

b_{79}	$b_{78} \quad \ldots \quad b_{64}$	$b_{63} \quad \ldots \quad b_0$
signe	exposant	mantisse

- Précision relative : $Pr \approx 2^{-64} \approx 5,4 \times 10^{-20}$
- Plus petit nombre codé (dénormalisé) :
  $exposant = 0$, $mantisse = 2^{-64}$,
  valeur $= 2^{-64} \times 2^{-16383} \approx 9,1.10^{-4952}$
- Plus grand nombre codé :
  $mantisse = 1,1...1 = 2 - 2^{-64} \approx 2$, $exposant = 16384$
  valeur $\approx 2 \times 2^{16384} = 2^{16385} \approx \ 2,8 \times 10^{4932}$

Exemple ilustratif - Simple précision

1. Code de 1
   - le bit de signe vaut 0
   - la mantisse vaut 0
   - l'exposant vaut 0+127 = 127
   - d'où le code [0 01111111 0..0]
2. code de $-12,125$
   - le bit de signe vaut 1
   - Retour à un codage de la forme $1, mantisse$
     $$12,125 \times 2^{-3} = 1,515625 \ (exposant = 127 + 3 = 130)$$
   - code de l'exposant 130 = 10000010
   - code de 1.515625 = $1,100001$ d'où $mantisse = 1000010...0$
   - d'où enfin le code [1 10000010 100010...0]

Exemple ilustratif - Double précision

1. Code de 124,356
   - Le bit de signe vaut 0
   - Retour à un codage de la forme $1, mantisse$
     $$124,356 \times 2^{-6} = 1,9430625$$
     $(exposant = 1023 + 6 = 1029)$
   - Code de l'exposant $1029 = 10000000101$
   - Code de la $mantisse = 1,9430625 =$
     $1,1111000101101100100010110100001110010101100000010000$
   - D'où le code qui tient difficilement sur la page ! ! !

     0 10000000101 1111000101101100100010110100001110010101100000010000

     Je compte sur vous pour vérifier !

Exemple ilustratif - Simple précision - Décodage

1. Soit le code $5C000000_{16}$ à décoder en simple précision
   - Code Binaire

5	$C(12)$	0	0	0	0	0	0
0101	1100	0000	0000	0000	0000	0000	0000

   - Bit de signe =0 : nombre positif
   - Code exposant = 10111000
   - Valeur exposant biaisé : 184, d'où exposant =184-127=57
   - Code de la *mantisse* = 1,0 (valeur 1)
   - D'où la valeur du code $2^{57} \approx 1.44 \times 10^{17}$