Codage des nombres réels

Il existe deux codages à virgule en machine : le codage en virgule fixe et le codage en virgule flottante (norme IEEE-754) existant en simple, double ou quadruple précision.
Le codage en virgule fixe est encore utilisé dans certains microcontrôleurs. Le codage en virgule flottante est utilisé partout ailleurs (ordinateurs, smartphones ...).

1. Le codage à virgule fixe.

Le principe du codage en virgule fixe est de retenir un nombre fixe de chiffres après la virgule.
Exemple :
Le nombre binaire $1011\ 1101_{(2)}$ codé en virgule fixe sur 4 bits est donc composé de 4 bits pour la partie entière (puissances positives de 2) du nombre et 4 bits pour la partie située après la virgule (puissances de 2 négatives). On a alors :

1	0	1	1		1	0	0	1
2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
Partie entière du nombre				**,**	**Partie décimale du nombre**			

$$10111001_{(2)}=1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+0\times2^{-3}+1\times2^{-4}=11{,}5625$$

2.Le codage en virgule flottante

Le codage à virgule fixe n'est pas pertinent pour l'ensemble des nombres. En effet les petits nombres demandent beaucoup de chiffres après la virgule alors que pour les grands nombres, c'est plutôt avant la virgule qu'il faut réserver de la place.
Ainsi a été défini le principe de virgule flottante (mobile) pour s'adapter à tous les nombres à virgule.
En informatique, pour représenter les nombres à virgule, on utilise une représentation similaire à la «notation scientifique» des calculatrices, sauf qu'elle est en base deux et non en base dix. Un tel nombre est représenté sous la forme :

$$(-1)^S\times(1+m)\times2^n$$

S : Signe du nombre
m : mantisse du nombre exprimée en puissances négatives de 2 (m < 1). La somme 1+m est donc comprise entre 1 inclus et 2 exclus.
n : Exposant du nombre (entier décalé)

Les nombres à virgule flottante doivent respecter une forme normalisée, afin que la représentation ne varie pas d'un matériel à l'autre. La norme standard est la norme IEEE 754. Elle définit 4 formats pour représenter des nombres à virgule flottante : simple précision (32 bits), la double précision (64 bits) et la double précision étendue (80 bits). Le format le plus courant est le format double précision, dont l'organisation des bits est le suivant :

b_{63}	b_{62}	b_{61}	...	b_{53}	b_{52}	b_{51}	b_{50}	b_{49}	...	b_2	b_1	b_0
Signe	**Exposant décalé**					**Mantisse**						

Elément	Nb de bits	Description du format <u>double précision</u>
Signe	1	**0** : Nombre positif ; **1** : Nombre négatif
Exposant décalé	11	L'exposant **n** est un entier relatif compris entre –1 022 et 1 023. Or il doit être représenté comme un entier naturel. Pour cela on utilise un décalage de 1023. L'exposant décalé possède donc une valeur comprise entre 1 et 2046. Les deux entiers 0 et 2 047 sont réservés pour certaines situations : NaN (Not a Number) ; +∞ ; −∞.
Décimales de la mantisse	52	Il s'agit d'un nombre binaire à virgule compris entre 1 inclus et 2 exclu comprenant 52 chiffres après la virgule. Il y a toujours un seul chiffre avant la virgule et ce chiffre est toujours un 1 ; il est donc inutile de le représenter. Les 52 bits pour représenter les 52 chiffres après la virgule.

<u>Exemple → Valeur du nombre flottant en double précision $C4693C3800000000</u>

$(C4693C3800000000)_{16} = (1100010001101001001111000011100000000000000000000000000000000 0000)_2$

$N = (1\ 10001000110\ 1001001111000011100000000000000000000000000000000000)_2$

S = 1 : le nombre est **négatif**.

Exposant décalé = $(10001000110)_2$
= 1094
donc l'exposant n = 1094 – 1023
n = 71.

La valeur de mantisse est égale à **1** virgule suivie des 52 bits de m :

$1,b_{51}\ b_{50}\ b_{49}\ ...b_2\ b_1\ b_0$

Mantisse=$(1,1001001111000011100000000000000000000000000000000000)_2$
$= 2^0 + 2^{-1} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-15} + 2^{-16} + 2^{-17}$
= 1,57720184326171875.

Résultat

$N = -1S \times (1+m) \times 2^n = -1^1 \times \mathbf{1,57720184326171875} \times 2^{\mathbf{71}} = -3,724062560669682106368 \times 10^{21}$

$(C4693C3800000000)_{16} = -3,724062560669682106368 \times 10^{21}$

Concernant les formats, la différence entre les encodages est le nombre alloué à la mantisse et à l'exposant. Le tableau suivant résume les différents encodages :

	Exposant (n)	Mantisse (m)	Valeur max
simple précision : 32 bits	8 bits (-126 à 127)	23 bits	≈3,402 823 5 × 10^{38}
Double précision : 64 bits	11 bits (-1022 à 1023)	52 bits	≈1,7976931348623157 × 10^{308}
Double précision étendu : 80 bits	15 bits (-16382 à 16383)	63 bits	≈1,1618471634347966 455915618 × 10^{4929}

Valeurs spéciales

Le format des nombres flottants ne permet pas de représenter le nombre 0. Ainsi des valeurs spécifiques sont réservées pour représenter 0, +∞, -∞. Par exemple dans le cas d'une simple précision (32 bits) l'encodage des valeurs spéciales est :

Signe	Exposant	Mantisse	Valeur spéciale
0	0	0	+0
1	0	0	-0
0	255	0	+∞
1	255	0	-∞
0	255	≠0	NaN

3.Les flottants en Python

Ils sont représentés selon la norme IEEE 754 double précision (format 64 bits). On peut utiliser la notation décimale ou scientifique pour le définir :

```
>>> x = 1.6
>>> 1.2e-4
0.00012
```

La fonction float permet de convertir un entier en flottant :

```
>>> float(-4)
-4.0
>>> int(5.9)
5
```

Les opérations arithmétiques usuelles s'appliquent également aux flottants. A noter que l'opérateur de division / produit toujours un résultat flottant.

```
>>> 3.4 + 0.012
3.412
>>> 1.2 / 2
0.6
>>> 4 / 2
2.0
```

Certaines expressions peuvent générer des valeurs spéciales, comme inf ou nan.

```
>>> x = 1e200
>>> x * x
inf
>>> (x * x) * 0
nan
```

Propriétés des flottants

Il faut être prudent lorsque l'on manipule des nombres flottants, en effet il ne faut jamais oublier que les calculs sur des flottants sont inexacts

```
>>> 1.2 * 3
2.5999999999999996
>>> 1.6 + (3.2 + 1.7)
6.5
>>> (1.6 + 3.2) + 1.7
6.500000000000001
>>> 1.5 * (3.2 + 1.4)
6.8999999999999995
>>> 1.5 * 3.2 + 1.5 * 1.4
6.9
>>> 0.1 + 0.2 == 0.3
False
```

A retenir

Les nombres flottants sont une **représentation approximative** des nombres réels dans un ordinateur. Une norme internationale (IEE754) définit définit un encodage en simple (32bits) ou double précision (64 bits).

Les opérations sur des flottants n'ont pas toujours les mêmes propriétés que sur les réels.