Chapitre 1

Systèmes de numération

Pour pouvoir utiliser et manipuler des nombres, il faut pouvoir en donner une représentation. Les chiffres qui sont utilisés aujourd'hui sont une invention relativement récente; en effet, les chiffres romains que l'on rencontre encore parfois nous rappellent qu'à l'époque romaine la représentation des nombres était différente. Notre système décimal, y compris l'invention du nombre zéro, est originaire d'Inde et daterait du cinquième ou sixième siècle de notre ère. Il s'est propagé dans le monde islamique et est arrivé en Europe entre le dixième et le douzième siècle à la suite de la traduction et de la diffusion des travaux des mathématiciens arabes (l'ouvrage *Liber abaci* de Fibonacci déjà présente ce système). Les chiffres que nous utilisons, 0, 1, 2, ..., 9, sont justement appelés chiffres arabes en raison de cette origine.

Les systèmes destinés à représenter les nombres sont appelés systèmes de numération et on peut en distinguer deux types différents. Le premier type ne tient pas systématiquement compte de la position des symboles utilisés et demande régulièrement l'introduction de nouveaux symboles; le système de numération basé sur les chiffres romains est de ce type. Il en va de même pour les anciens systèmes égyptien, chinois, grec et hébreux. Le second type regroupe les systèmes dits positionnels (ou à position) comme notre système décimal usuel pour lesquels le nombre de symboles (chiffres) utilisés est fixé, ce qui rend leur emploi facile. Un autre exemple d'un tel système (bien qu'incomplet, car sans zéro) est le système babylonien de base 60 qui est à l'origine de la division des heures et minutes en 60 parties ainsi que de la division de la circonférence en $360 = 6 \cdot 60$ définissant l'unité de degré pour la mesure des angles.

Avec l'avènement de l'informatique, les systèmes binaire (base 2), octal (base 8) et hexadécimal (base 16), qui sont des systèmes positionnels, revêtent une importance toute particulière et apparaîtront fréquemment au cours de ce chapitre.

1.1 Systèmes positionnels

Définition 1.1. Un système de numération est un système de représentation des nombres à l'aide d'une suite de symboles appelés chiffres.

Définition 1.2. Un système de numération positionnel est un système de numération dans lequel on utilise un nombre fini de chiffres et la valeur d'un chiffre dépend de la position qu'il occupe dans la suite des chiffres représentant le nombre.

***Exemple* 1.3.**
Le système décimal usuel utilise les chiffres 0, 1, ..., 9. Par exemple, le nombre 971 est composé de trois chiffres dont la valeur dépend de leur position; en effet le 9 figurant en tête du nombre représente neuf centaines, alors que dans le nombre 179, le chiffre 9 ne représente que neuf unités.

1.2 Représentation des entiers en base b

Définition 1.4. La base b d'un système de numération positionnel est le nombre de chiffres utilisés pour représenter les nombres dans ce système. Notation: le nombre x représenté en base b est noté $(x)_b$.

Un entier x en base b est constitué par une suite de chiffres $a_n, a_{n-1}, \ldots, a_0$, juxtaposés que l'on note

$$(x)_b = (a_n a_{n-1} a_{n-2} \ldots a_0)_b \,, \tag{1.1}$$

où les a_i, $0 \leqslant i \leqslant n$, sont les chiffres utilisés en base b avec $0 \leqslant a_i \leqslant b-1$, et la valeur de l'indice i est associé à la position du chiffre dans le nombre.

***Exemple* 1.5.**
Dans le nombre $(417)_{10}$*, on a* $a_2 = 4$*,* $a_1 = 1$*,* $a_0 = 7$*.*

En fait, à la notation de l'équation (1.1) correspond la représentation polynomiale suivante,

$$\begin{aligned}(x)_b &= (a_n a_{n-1} a_{n-2} \ldots a_0)_b \\ &= a_n b^n + a_{n-1} b^{n-1} + \ldots + a_1 b^1 + a_0 b^0 \,. \end{aligned} \tag{1.2}$$

Les chiffres représentant le nombre en base b sont les coefficients des puissances successives de b. Ceci est la caractéristique essentielle des systèmes de numération positionnels.

***Exemple* 1.6.**
Dans le nombre $(417)_{10}$*, on a* $a_2 = 4$*,* $a_1 = 1$*,* $a_0 = 7$*, et* 4 *est le coefficient de* $10^2 = 100$ *(centaines),* 1 *celui de* $10^1 = 10$ *(dizaines),* 7 *celui de* $10^0 = 1$ *(unités).*

Outre les bases 2 et 8, la base $b = 16$ se rencontre fréquemment en informatique[1]. Comme 16 chiffres sont nécessaires pour pouvoir l'utiliser, aux chiffres $0, 1, \ldots, 9$, on ajoute les lettres A, B, C, D, E, F, qui représentent respectivement les nombres de 10 à 15. Ainsi on dispose des 16 symboles nécessaires. On peut aussi simplement utiliser les nombres 10 à 15 au lieu des lettres A à F, mais dans ce cas on utilise un point-virgule pour séparer les "chiffres" dans l'écriture du nombre; par exemple, $(2A7B)_{16} = (2; 10; 7; 11)_{16}$. Pour les systèmes de base $b > 16$ (penser au système babylonien déjà mentionné pour lequel $b = 60$), on utilisera aussi la notation avec points-virgule.

1.3 Changements de base pour les nombres entiers

1.3.1 Cas général (de et vers la base 10)

Conversion de la base b en base 10

Pour convertir un nombre entier donné en base b dans la base 10, on utilise la représentation polynomiale (1.2) du nombre.

Procédure de conversion

1. Multiplier chaque chiffre du nombre par la puissance de la base correspondante à sa position dans le nombre.
2. Effectuer la somme des produits obtenus pour obtenir le nombre dans le système décimal.

***Exemple* 1.7.**
Convertir $(321)_5$ *en base* 10*.*

$$(321)_5 = 3 \cdot 5^2 + 2 \cdot 5^1 + 1 \cdot 5^0 = 75 + 10 + 1 = 86 \,.$$

[1] La principale raison pour laquelle on utilise les bases 8 et 16 est simplement parce qu'elles fournissent une écriture abrégée de la base 2, voir les sections 1.3.2 et 1.8.

***Exemple* 1.8.**
Convertir $(2;34;12)_{60}$ *en base* 10.

$$(2;34;12)_{60} = 2 \cdot 60^2 + 34 \cdot 60^1 + 12 \cdot 60^0 = 7200 + 2040 + 12 = 9252\,.$$

Conversion de la base 10 **en base** b

Pour convertir un nombre entier donné en base 10 en base b, il faut l'exprimer comme une somme de puissances de b, et les coefficients de ces puissances successives sont les chiffres du nombre représenté en base b.

***Exemple* 1.9.**
Convertir $(50)_{10}$ *en base* 2.

$$50 = 32 + 16 + 2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0\,.$$

Les coefficients de la représentation polynomiale sont $1, 1, 0, 0, 1, 0$.
Donc $(50)_{10} = (110010)_2$.

***Exemple* 1.10.**
Convertir $(195)_{10}$ *en base* 3.

$$195 = 2 \cdot 3^4 + 1 \cdot 3^3 + 0 \cdot 3^2 + 2 \cdot 3^1 + 0 \cdot 3^0\,.$$

Les coefficients de la représentation polynomiale sont $2, 1, 0, 2, 0$.
Donc $(195)_{10} = (21020)_3$.

Pour obtenir plus facilement la représentation d'un nombre x en base b, il suffit de remarquer que le dernier chiffre (à droite) de cette représentation est le reste de division de x par b. Donc pour obtenir la représentation en base b, il suffit de diviser x ainsi que chacun des quotients successifs obtenus par b jusqu'à ce qu'un quotient nul soit obtenu. La suite des restes de division dans l'ordre inverse de leur obtention donne la représentation du nombre en base b.

Procédure de conversion

1. Diviser le nombre à convertir par la base b (division entière).
2. Répéter la division par b avec le quotient obtenu à l'étape précédente jusqu'à ce le quotient soit nul.
3. La représentation du nombre en base b est obtenue en écrivant la suite des restes des divisions en commençant par le dernier.

***Exemple* 1.11.**
Conversion de $(50)_{10}$ *en base* 2.

$$\begin{array}{rcrcr} 50 & = & 2 \cdot 25 & + & 0 \\ 25 & = & 2 \cdot 12 & + & 1 \\ 12 & = & 2 \cdot 6 & + & 0 \\ 6 & = & 2 \cdot 3 & + & 0 \\ 3 & = & 2 \cdot 1 & + & 1 \\ 1 & = & 2 \cdot 0 & + & 1 \end{array} \quad , \quad \textit{donc } (50)_{10} = (110010)_2.$$

***Exemple* 1.12.**
Conversion de $(195)_{10}$ *en base* 3.

$$\begin{array}{rcrcr} 195 & = & 3 \cdot 65 & + & 0 \\ 65 & = & 3 \cdot 21 & + & 2 \\ 21 & = & 3 \cdot 7 & + & 0 \\ 7 & = & 3 \cdot 2 & + & 1 \\ 2 & = & 3 \cdot 0 & + & 2 \end{array} \quad , \quad \textit{donc } (195)_{10} = (21020)_3.$$

1.3.2 Cas particuliers

Conversion de la base b en base b^n

On partage le nombre à convertir en tranches de n chiffres en partant de la droite et on remplace chaque tranche par son équivalent en base b^n.

***Exemple* 1.13.**
Convertir $(2021120)_3$ *en base* $9 = 3^2$ *(*$b = 3$, $n = 2$*).*

$$\underbrace{2}_{2}\,\underbrace{02}_{2}\,\underbrace{11}_{4}\,\underbrace{20}_{6}$$

donc $(2021120)_3 = (2246)_9$.

***Exemple* 1.14.**
Convertir $(10110111001)_2$ *en base* $8 = 2^3$ *(*$b = 2$, $n = 3$*).*

$$\underbrace{10}_{2}\,\underbrace{110}_{6}\,\underbrace{111}_{7}\,\underbrace{001}_{1}$$

donc $(10110111001)_2 = (2671)_8$.

Conversion de la base b^n en base b

On remplace chaque chiffre du nombre par son équivalent en base b (écriture sur n positions).

***Exemple* 1.15.**
Convertir $(2F65)_{16}$ *en base* 2 *(donc* $b = 2$, $n = 4$*).*

$$\underbrace{2}_{0010}\,\underbrace{F}_{1111}\,\underbrace{6}_{0110}\,\underbrace{5}_{0101}$$

donc $(2F65)_{16} = (10111101100101)_2$.

1.4 Opérations arithmétiques en base b

Les opérations arithmétiques (addition, soustraction, multiplication, division) entre nombres entiers représentés en base b s'effectuent d'une façon analogue à celles en base 10, hormis évidemment le fait qu'il faille utiliser correctement les opérations dans la base considérée, en particulier les tables de multiplication[2].

1.4.1 Addition

Dans tout système de numération positionnel, les symboles sont utilisés de façon cyclique et la longueur du cycle correspond à la base du système de numération.

Pour additionner deux nombres, on les superpose en colonnes en alignant les chiffres de même position. On effectue alors la somme des chiffres par colonne, à partir de la colonne la plus à droite (celle des unités).

***Exemple* 1.16.**
Additionner en base 10 *les nombres* 243 *et* 92.
On superpose les nombres à additionner en colonnes.
Somme des chiffres dans la colonne des unités: $3 + 2 = 5$.
Puis dans la colonne des dizaines: $4 + 9 = 13 = 10 + 3$*; on indique* 3 *dans cette colonne et on retient* 1 *pour la colonne des centaines.*
Finalement, dans la colonne des centaines $2 + 1 = 3$.

$$\begin{array}{rr} & 243 \\ + & 92 \\ \hline & 335 \end{array}$$

[2]Par exemple, en base 4, on a $2 \cdot 3 = 12$.

***Exemple* 1.17.**
Addition en base 2 *de* $(1111)_2$ *et* $(110)_2$*. Il faut juste se rappeler que dans cette base on a* $1 + 1 = 10$*.*
Colonne des unités: $1 + 0 = 1$*.*
Deuxième colonne: $1 + 1 = 10$*; retenir* 1 *pour la colonne suivante.*
Troisième colonne: $1 + 1 + 1 = 11$*; retenir* 1 *pour la colonne suivante.*
Quatrième colonne: $1 + 1 = 10$*; retenir* 1 *pour la colonne suivante.*
Cinquième colonne: $0 + 1 = 1$*.*

$$\begin{array}{rr} & 1111 \\ + & 110 \\ \hline & 10101 \end{array}$$

1.4.2 Soustraction

Pour soustraire deux nombres, on les superpose en colonnes en alignant les chiffres de même position. On soustrait alors les chiffres colonne par colonne à partir de la colonne la plus à droite (celle des unités). Si le chiffre sur la ligne du haut est supérieur ou égal à celui de la ligne du bas, on effectue simplement la soustraction. Si le chiffre sur la ligne du haut est inférieur à celui de la ligne du bas, on effectue d'abord "un emprunt" pour pouvoir soustraire en ajoutant 10 au chiffre du haut et en ajoutant 1 au chiffre du bas dans la colonne suivante. Ensuite on effectue la soustraction du chiffre du haut (augmenté de 10) et du chiffre en dessous.

***Exemple* 1.18.**
En base 10*, soustraire* 244 *de* 525*.*
Colonne des unités: $5 - 4 = 1$*.*
Colonne des dizaines: comme $2 < 4$*, on "emprunte"* 10 *et* $12 - 4 = 8$*.*
Colonne des centaines: $5 - (2 + 1) = 2$ *(on "restitue l'emprunt" en ajoutant* 1 *à soustraire).*

$$\begin{array}{rr} & 525 \\ - & 244 \\ \hline & 281 \end{array}$$

***Exemple* 1.19.**
En base 2*, soustraire* 110 *de* 10010*.*
Colonne des unités: $0 - 0 = 0$*.*
Deuxième colonne: $1 - 1 = 0$*.*
Troisième colonne: $10 - 1 = 1$*; retenir* 1 *à soustraire pour la colonne suivante.*
Quatrième colonne: $10 - 1 = 1$*; retenir* 1 *à soustraire pour la colonne suivante.*
Cinquième colonne: $1 - 1 = 0$*.*

$$\begin{array}{rr} & 10010 \\ - & 110 \\ \hline & 1100 \end{array}$$

***Exemple* 1.20.**
En base 5*, soustraire* 241 *de* 432*.*
Colonne des unités: $2 - 1 = 1$*.*
Deuxième colonne: $13 - 4 = (8)_{10} - (4)_{10} = 4$*;*
on retient 1 *à soustraire pour la colonne suivante.*
Troisième colonne: $4 - (2 + 1) = 1$*.*

$$\begin{array}{rr} & 432 \\ - & 241 \\ \hline & 141 \end{array}$$

1.4.3 Multiplication

Pour multiplier deux nombres, on les superpose en colonnes en alignant les chiffres de même position. On multiplie ensuite consécutivement le nombre sur la ligne du haut par chaque chiffre du nombre de la ligne du bas en partant de la droite et en décalant d'une colonne à chaque multiplication. C'est là la difficulté de cette opération, car les tables de multiplication sont évidemment différentes en base b et en base 10; or les seules tables que que l'on connaisse par cœur sont celles en base 10. Finalement, on additionne les produits calculés pour aboutir au résultat.

***Exemple* 1.21.**
Multiplier en base 10 *les nombres* 531 *et* 25*.*

$531 \cdot 5 = 2655$,
$531 \cdot 2 = 1062$,
puis on additionne (en tenant compte du décalage).

$$\begin{array}{rr} & 531 \\ \times & 25 \\ \hline & 2655 \\ & 1062 \\ \hline & 13275 \end{array}$$

***Exemple* 1.22.**
Multiplier en base 2 *les nombres* 10110 *et* 11.

$10110 \cdot 1 = 10110$,
puis on additionne (en tenant compte du décalage).

$$\begin{array}{rr} & 10110 \\ \times & 11 \\ \hline & 10110 \\ & 10110 \\ \hline & 1000010 \end{array}$$

***Exemple* 1.23.**
Multiplier en base 6 *les nombres* 5432 *et* 12.

$5432 \cdot 2 = 15304$,
$5432 \cdot 1 = 5432$,
puis on additionne (en tenant compte du décalage).

$$\begin{array}{rr} & 5432 \\ \times & 12 \\ \hline & 15304 \\ & 5432 \\ \hline & 114024 \end{array}$$

1.4.4 Division entière

A nouveau, comme pour les autres opérations, la division entière en base b s'effectue d'une façon analogue à la division en base 10. Suivons l'exemple 1.24 ci-dessous en base 10 pour rappeler la méthode. Dans le dividende 3573, on prend autant de chiffres que néccesaire en partant de la gauche pour obtenir un nombre supérieur ou égal au diviseur 25; il faut donc prendre ici 35. On détermine ensuite le plus grand multiple de 25 inférieur ou égal à 35, soit dans ce cas 1 que l'on écrit en dessous du diviseur 25. On effectue la multiplication de 1 par le diviseur 25 et on inscrit le résultat (dans notre cas 25) sous le dividende, puis on le soustrait du nombre 35, ce qui donne 10. On abaisse ensuite le chiffre suivant du dividende, ce qui donne 107. On réitère alors le processus en déterminant le plus grand multiple de 25 inférieur ou égal à 107, soit dans ce cas 4, que l'on écrit en dessous du diviseur 25 à la droite du 1 obtenu précédemment, puis on multiplie 4 par 25 pour obtenir 100 que l'on inscrit en-dessous de 107 pour pouvoir le soustraire de ce dernier, et ainsi de suite. Le processus s'arrête lorsqu'il n'est plus possible de former un nombre à diviser supérieur ou égal au diviseur (il n'y a alors plus de chiffre à abaisser du dividende). Le dernier nombre restant est le reste de la division, le nombre formé sous le diviseur est le quotient. Dans notre exemple, le quotient est 142, le reste 23.

***Exemple* 1.24.**

Diviser en base 10 *le nombre* 3573 *par* 25.
Résultat: 142 *reste* 23.

$$\begin{array}{r|l} 3573 & 25 \\ \hline \underline{-25} & 142 \\ 107 & \\ \underline{-100} & \\ 73 & \\ \underline{-50} & \\ 23 & \end{array}$$

***Exemple* 1.25.**

Diviser en base 2 *le nombre* 11101 *par* 10.
Résultat: $(1110)_2$ *reste* $(1)_2$.

$$\begin{array}{r|l} 11101 & 10 \\ \hline \underline{-10} & 1110 \\ 11 & \\ \underline{-10} & \\ 10 & \\ \underline{-10} & \\ 01 & \end{array}$$

***Exemple* 1.26.**

Diviser en base 6 *le nombre* 3452 *par* 24.
Résultat: $(123)_6$ *reste* $(12)_6$.

3452	24
−24	123
105	
−52	
132	
−120	
12	

1.5 Représentation des nombres réels en base b

Pour simplifier l'exposé, nous présentons le sujet avec des nombres réels dont la représentation ne nécessite qu'un nombre fini de chiffres (pour la partie fractionnaire), mais les résultats et calculs que nous allons introduire restent valides pour des nombres n'ayant pas un nombre fini de chiffres dans leur représentation[3].

Un nombre réel x représenté en base b est constitué par une suite de chiffres $a_n, a_{n-1}, \ldots, a_0, a_{-1}, \ldots, a_{-m}$ et

$$\begin{aligned}(x)_b &= (a_n; a_{n-1}; \ldots; a_0 \cdot a_{-1}; \ldots; a_{-m})_b, \\ &= a_n b^n + \ldots + a_0 b^0 + a_{-1} b^{-1} + \ldots + a_{-m} b^{-m} \end{aligned} \tag{1.3}$$

où les coefficients a_i, $-m \leqslant i \leqslant n$, sont des chiffres de la base b ($0 \leqslant a_i \leqslant b-1$) et la valeur de l'indice i est associé à la position du chiffre dans la représentation du nombre. Lorsque l'indice est positif, il correspond à une puissance positive, et s'il est négatif, il correspond à une puissance négative. Un point (ou une virgule) sépare les chiffres correspondant aux puissances positives ou nulle de ceux correspondant aux puissances négatives.

***Exemple* 1.27.**
En base 10, *si* $x = (25.13)_{10}$, *alors* $a_1 = 2$, $a_0 = 5$, $a_{-1} = 1$, $a_{-2} = 3$ *et*

$$\begin{aligned} x &= a_n \cdot 10^n + \ldots + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \ldots + a_{-m} \cdot 10^{-m} \\ &= 2 \cdot 10^1 + 5 \cdot 10^0 + 1 \cdot 10^{-1} + 3 \cdot 10^{-2} \end{aligned}$$

1.6 Changements de base pour les nombres réels

Nous avons vu à la section 1.3 comment procéder pour les changements de base avec des nombres entiers, il suffit d'adapter les procédures de conversion pour les nombres non entiers.

1.6.1 Cas général (de et vers la base 10)

Conversion de la base b en base 10

Pour convertir un nombre non entier donné en base b dans la base 10, on utilise la représentation polynomiale (1.3) du nombre.

Procédure de conversion

1. Multiplier chaque chiffre du nombre par la puissance de la base correspondante à sa position dans le nombre.

2. Effectuer la somme des produits obtenus pour obtenir le nombre dans le système décimal.

[3] Avoir une représentation avec un nombre de chiffres fini ou non dépend de la base. En effet, par exemple $(0.1)_{10} = (0.0001100110011\ldots)_2$.

***Exemple* 1.28.**
En base 8, *si* $x = (12.56)_8$, *alors* $a_1 = 1$, $a_0 = 2$, $a_{-1} = 5$, $a_{-2} = 6$ *et*

$$\begin{aligned} x &= a_1 \cdot 8^1 + a_0 \cdot 8^0 + a_{-1} \cdot 8^{-1} + a_{-2} \cdot 8^{-2} \\ &= 1 \cdot 8^1 + 2 \cdot 8^0 + 5 \cdot 8^{-1} + 6 \cdot 8^{-2} \\ &= 10.71875 \end{aligned}$$

Conversion de la base 10 **en base** b

Il faut traiter séparément la partie entière et la partie fractionnaire. On utilise la procédure de la section 1.3.1 pour la partie entière. Pour la partie fractionnaire, on multiplie successivement par b en ne conservant que la partie entière du résultat. La suite de ces parties entières (transcrite de haut en bas) donne la représentation en base b de la partie fractionnaire du nombre à convertir.

Procédure de conversion

1. Traiter la partie entière du nombre à convertir.
2. Multiplier la partie fractionnaire par b, isoler la partie entière du résultat.
3. Répéter la multiplication par b sur la partie fractionnaire du précédent résultat.
4. La représentation du nombre en base b est obtenue en écrivant la partie entière convertie, le point (ou la virgule), et la suite des parties entières obtenues par les multiplications ci-dessus. Cette suite peut ne pas être finie.

***Exemple* 1.29.**
Convertir $(4.375)_{10}$ *en base* 4. *On traite d'abord la partie entière:* $(4)_{10} = (10)_4$. *Ensuite, on considère la partie fractionnaire* $(0.375)_{10}$ *que l'on multiplie successivement par* b *et on ne retient que les parties entières de ces multiplications.*

$0.375 \cdot 4$	$= 1.5$	*partie entière*	1
$0.5 \cdot 4$	$= 2.0$		2

Donc $(4.375)_{10} = (10.12)_4$

***Exemple* 1.30.**
Convertir $(5.1)_{10}$ *en base* 2. *On traite d'abord la partie entière:* $(5)_{10} = (101)_2$. *Ensuite, on considère la partie fractionnaire* $(0.1)_{10}$ *que l'on multiplie successivement par* b *et on ne retient que les parties entières de ces multiplications.*

$0.1 \cdot 2$	$= 0.2$	*partie entière*	0
$0.2 \cdot 2$	$= 0.4$		0
$0.4 \cdot 2$	$= 0.8$		0
$0.8 \cdot 2$	$= 1.6$		1
$0.6 \cdot 2$	$= 1.2$		1
$0.2 \cdot 2$	$= 0.4$		0
	$\vdots$		

On voit que la suite des parties entières se répète, puisque la dernière ligne est identique à la seconde. Le développement fractionnaire est donc périodique (et la période est 0011*). La suite des parties entières ci-dessus est* $0, 0, 0, 1, 1, 0, 0, 1, 1, \ldots$ *Donc* $(5.1)_{10} = (101.000110011\ldots)_2 = (101.0\overline{0011})_2$ *(on indique la période en la surlignant).*

1.6.2 Cas particuliers

Conversion de la base b **en base** b^n

On procède comme à la section 1.3.2 en traitant séparément la partie entière et la partie fractionnaire. La seule différence est que pour la partie fractionnaire il faut former des tranches

en partant de la gauche. Autrement dit, les tranches de n chiffres sont toujours à former en partant du point (ou de la virgule) décimal, et l'écriture doit toujours se faire sur n chiffres (rajouter des zéros au besoin à l'extrême droite et à l'extrême gauche du nombre pour former les tranches).

***Exemple* 1.31.**
Convertir $(1110.01010111)_2$ *en base* $8 = 2^3$ *(*$b = 2$, $n = 3$*).*

$$\underbrace{001}_{1}\underbrace{110}_{6} \cdot \underbrace{010}_{2}\underbrace{101}_{5}\underbrace{110}_{6}$$

donc $(1110.01010111)_2 = (16.256)_8$.

Conversion de la base b^n en base b

On procède comme à la section 1.3.2 en distinguant à nouveau les parties entière et fractionnaire. On représente chaque chiffre de base b^n en un nombre de base b avec une écriture sur n positions.

***Exemple* 1.32.**
Convertir $(4B.3E2)_{16}$ *en base* 4 *(*$b = 4$, $n = 2$*).*

$$\underbrace{4}_{10}\underbrace{B}_{23} \cdot \underbrace{3}_{03}\underbrace{E}_{32}\underbrace{2}_{02}$$

donc $(4B.3E2)_{16} = (1023.033202)_4$.

1.7 Nombres réels périodiques

1.7.1 Cas général (de et vers la base 10)

Conversion de la base b en base 10

Pour convertir un nombre réel périodique de base b en base 10, on fait appel à la proposition suivante.

Proposition 1.33. *Soient b un entier, $b > 1$, et β un entier tel que $1 < \beta \leqslant b - 1$. Alors les relations suivantes sont vérifiées.*

$$1) \qquad (0.\overline{1})_b = \frac{1}{b-1}$$
$$2) \qquad (0.\overline{\beta})_b = \frac{\beta}{b-1}$$

Démonstration. 1) Notons x le nombre représenté par $(0.\overline{1})_b$, $x = (0.\overline{1})_b$. On peut donc écrire

$$\begin{aligned} b \cdot x &= (10)_b \cdot (0.\overline{1})_b \\ &= (1.\overline{1})_b \\ &= (1)_b + (0.\overline{1})_b \\ &= 1 + x\,. \end{aligned}$$

On obtient ainsi l'équation

$$b \cdot x = 1 + x$$

donc $x = \dfrac{1}{b-1}$.
2) En utilisant la démonstration du point 1), on a

$$\begin{aligned}(0.\overline{\beta})_b &= \beta \cdot (0.\overline{1})_b \\ &= \beta \cdot \frac{1}{b-1} \\ &= \frac{\beta}{b-1}\,.\end{aligned}$$

□

Cette proposition s'applique aux nombres réels périodiques ayant une période comportant un seul chiffre en base b et dont la période commence juste après la virgule. Il faut donc se ramener à cette situation en appliquant les deux procédures de réduction ci-dessous si besoin est.

1. Si la période du nombre considéré en base b commence au $(n+1)$–ème chiffre après la virgule, alors il faut multiplier le nombre par b^n afin que la période commence juste après la virgule.

2. Si la période du nombre considéré en base b comporte n chiffres, il faut convertir ce nombre en base b^n afin d'obtenir un nombre dont la période est constituée par un seul chiffre.

Après application au besoin de 1. ou 2. ci-dessus, on est en mesure d'utiliser la proposition. Si on a appliqué 1., il faudra bien sûr pour donner le résultat final diviser par b^n.

***Exemple* 1.34.**
Convertissons en base 10 *le nombre* $(1220.1\overline{102})_3$.
1. On déplace la période juste après la virgule en multipliant par 3,

$$(1220.1\overline{102})_3 \cdot 3 = (12201.\overline{102})_3$$

2. Comme la période comporte 3 chiffres, on convertit le nombre en base $3^3 = 27$ *afin d'avoir une période à un seul chiffre.*

$$\underbrace{12}_{5}\underbrace{201}_{19} \cdot \underbrace{102}_{11}\underbrace{102}_{11}\ldots$$

Ainsi, en appliquant la proposition,

$$\begin{aligned}(12201.\overline{102})_3 &= (5; 19.\overline{11})_{27} \\ &= 5 \cdot 27^1 + 19 \cdot 27^0 + \frac{11}{26} \\ &= 154 + \frac{11}{26}.\end{aligned}$$

Pour obtenir le résultat final, il faut diviser par 3 *pour compenser la multiplication du point 1. ci-dessus, d'où la réponse* $(1220.1\overline{102})_3 = \dfrac{154}{3} + \dfrac{11}{26 \cdot 3} = 51.47436\ldots$

Conversion de la base 10 **en base** b

Ce type de conversion est tout à fait similaire à la conversion étudiée précédemment pour un nombre réel quelconque (voir 1.6.1).

***Exemple* 1.35.**
Convertir $(0.\overline{2})_{10}$ *en base* 3.

$$\begin{array}{lll} 0.\overline{2}\cdot 3 & = 0.\overline{6} & \textit{partie entière} \quad 0 \\ 0.\overline{6}\cdot 3 & = 1.\overline{9} & 1 \\ 0.\overline{9}\cdot 3 & = 2.\overline{9} & 2 \\ 0.\overline{9}\cdot 3 & = 2.\overline{9} & 2 \\ & \vdots & \vdots \end{array}$$

D'où la réponse $(0.01\overline{2})_3$.

1.7.2 Cas particuliers

Conversion de la base b en base b^n

Ce type de conversion s'effectue selon la même méthode que celle qui s'applique à un nombre réel quelconque (voir 1.6.2). Cependant, il convient d'ajuster la méthode selon la longueur de la période et sa position par rapport à la virgule (point décimal) en utilisant les procédures de réduction décrites à la section 1.7.1.

***Exemple* 1.36.**
Conversion de $(0.01\overline{21})_3$ *en base* 27.
Il faut former des tranches de 3 *chiffres pour convertir en base* 27 *(puisque* $3^3 = 27$*) et ceci y compris pour la période. On écrit donc*

$$\begin{aligned} (0.01\overline{21})_3 &= (0.012\overline{12})_3 \\ &= (0.012\,\overline{121\,212})_3 \end{aligned}$$

et

$$\begin{aligned} (012)_3 &= 5 \\ (121)_3 &= 16 \\ (212)_3 &= 23 \end{aligned}$$

d'où $(0.01\overline{21})_3 = (0.5; \overline{16; 23})_{27}$.

Conversion de la base b^n en base b

Ce type de conversion s'effectue selon le même procédé que pour un nombre réel quelconque (voir 1.6.2).

***Exemple* 1.37.**
Conversion de $(0.5; \overline{8; 11})_{16}$ *en base* 4. *On a*

$$\begin{aligned} 5 &= (1; 1)_4 \\ 8 &= (2; 0)_4 \\ 11 &= (2; 3)_4 \end{aligned}$$

donc $(0.5; \overline{8; 11})_{16} = (0.11\overline{2023})_4$.

1.8 Les systèmes binaire, octal et hexadécimal

Comme il a déjà été mentionné, une attention toute particulière doit être portée au système binaire puisque celui-ci est au cœur des ordinateurs. Cependant, le système binaire présente deux inconvénients majeurs par rapport au système décimal usuel. L'écriture est bien plus compacte dans le système décimal et le peu de symboles dans le système binaire (0 et 1) rend sa lecture difficile. Ainsi, dans le but de se rapprocher du système décimal, on utilise les système octal (base 8) et hexadécimal (base 16). En effet, tous deux sont des systèmes dont les bases

octal	binaire
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

hexadécimal	binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

TAB. 1.1 – Tableaux de conversion octal-binaire et hexadécimal-binaire.

respectives sont des puissances de 2 ($8 = 2^3$ et $16 = 2^4$), donc il est aisé de passer de la base 2 à l'une de ces autres bases (et vice-versa), et l'écriture dans ces bases devient plus compacte. Le système hexadécimal présente une petite particularité puisqu'il utilise outre les chiffres 0 à 9 les lettres A à F comme chiffres pour représenter les nombres de 10 à 15, $A = (10)_{10}$, $B = (11)_{10}$, $C = (12)_{10}$, $D = (13)_{10}$, $E = (14)_{10}$, $F = (15)_{10}$. Finalement rappelons que pour convertir un nombre binaire en octal, on prend les chiffres par tranches de 3 en partant de la part et d'autre du point décimal (ou virgule) et pour convertir un nombre binaire en hexadécimal, on forme des tranches de 4 chiffres. Les tables 1.1 indiquent la conversion des chiffres des systèmes octal et hexadécimal en binaire.

1.9 La complémentation

La complémentation est un méthode permettant de soustraire deux nombres sans utiliser l'algorithme classique de la soustraction, mais en utilisant l'addition. La motivation pour l'introduction d'une telle méthode est de pouvoir utiliser dans le processeur le même circuit électronique à la fois pour l'addition et la soustraction.

Définition 1.38. On appelle complément à $(b-1)$ d'un nombre $(x)_b$ en base b le nombre $(\overline{x})_b$ obtenu en soustrayant à $(b-1)$ chaque chiffre de $(x)_b$.

***Exemple* 1.39.**

Calculons le complément à 9 *de* $(153)_{10}$*:*

$$\begin{array}{rr} & 999 \\ - & 153 \\ \hline & 846 \end{array}$$

***Exemple* 1.40.**

Calculons le complément à 5 *de* $(243)_6$*:*

$$\begin{array}{rr} & 555 \\ - & 243 \\ \hline & 312 \end{array}$$

***Exemple* 1.41.**

Calculons le complément à 1 *de* $(10101)_2$*:*

$$\begin{array}{rr} & 11111 \\ - & 10101 \\ \hline & 01010 \end{array}$$

Définition 1.42. On appelle complément à b d'un nombre $(x)_b$ en base b le complément à $(b-1)$ auquel on ajoute 1.

En reprenant les exemples précédents,

- $(847)_{10}$ est le complément à 10 de $(153)_{10}$
- $(313)_6$ est le complément à 6 de $(243)_6$
- $(01011)_2$ est le complément à 2 de $(10101)_2$

Décrivons la procédure à suivre pour soustraire deux nombres en utilisant la méthode de la complémentation. On souhaite effectuer $x - y$ où $x > y$ en base b.

1. On rajoute si nécessaire des zéros à gauche de y afin que x et y comportent le même nombre de chiffres.
2. On calcule $(\overline{y})_b$, le complément à $(b-1)$ de $(y)_b$.
3. On calcule le complément à b de $(y)_b$ (qui est simplement $(\overline{y})_b + 1$).
4. On effectue l'addition $(x)_b + (\overline{y})_b + 1$.
5. On supprime le chiffre 1 obtenu en tête du résultat au point 4.

***Exemple* 1.43.**
$x = (315)_{10}$, $y = (103)_{10}$. *Donc* $(\overline{y})_{10} = (896)_{10}$, *et*

$$\begin{array}{rr} & 315 \\ + & 896 \\ + & 1 \\ \hline & 1212 \end{array}$$

Le résultat est donc $(212)_{10}$.

***Exemple* 1.44.**
$x = (5024)_{10}$, $y = (37)_{10}$. *Donc* $(\overline{y})_{10} = (9962)_{10}$, *et*

$$\begin{array}{rr} & 5024 \\ + & 9962 \\ + & 1 \\ \hline & 14987 \end{array}$$

Le résultat est donc $(4986)_{10}$.

***Exemple* 1.45.**
$x = (431)_5$, $y = (12)_5$. *Donc* $(\overline{y})_5 = (432)_5$, *et*

$$\begin{array}{rr} & 431 \\ + & 432 \\ + & 1 \\ \hline & 1414 \end{array}$$

Le résultat est donc $(414)_5$.

***Exemple* 1.46.**
$x = (11001)_2$, $y = (1110)_2$. *Donc* $(\overline{y})_2 = (10001)_2$, *et*

$$\begin{array}{rr} & 11001 \\ + & 10001 \\ + & 1 \\ \hline & 101011 \end{array}$$

Le résultat est donc $(1011)_2$.

***Remarque* 1.47.** Lorsque l'on souhaite effectuer $x-y$ avec $x < y$, on transforme l'opération en $-(y-x)$.

***Exemple* 1.48.**
$x = (103)_{10}$, $y = (315)_{10}$, *et* $x - y = -(y - x) = -((315)_{10} - (103)_{10}) = -(212)_{10}$ *(cf. exemple 1.43 ci-dessus).*

Dans le système décimal, il n'est pas nécessaire d'utiliser le complémentaire pour soustraire. Cependant, lorsque la soustraction est effectuée par un circuit électronique, *i.e.* en binaire, la complémentation permet d'utiliser le même circuit pour l'addition et la soustraction. Or en base 2 obtenir le complément à 1 d'un nombre est une opération très simple puisqu'il suffit d'échanger les 0 en 1 et réciproquement.

1.9.1 La représentation des entiers négatifs

Il est possible d'utiliser les compléments pour représenter des entiers négatifs. Par exemple, $-(324)_5$ peut être représenté par le complément de $(324)_5$ qui est $(121)_5$. Cependant, cette façon de faire n'est pas univoque. En effet, comment savoir si on se réfère au nombre $(121)_5$ ou $-(324)_5$? Afin de préciser le cadre dans lequel on travaille on normalise l'écriture, c'est–à–dire:

1. fixer le nombre de chiffres intervenant dans l'écriture du nombre,
2. fixer une convention (s'appliquant généralement sur le premier chiffre du nombre) indiquant s'il s'agit d'un nombre positif ou négatif.

***Exemple* 1.49.**
Considérons une écriture sur 8 *chiffres en base* 6. *Les entiers commençant par* $0, 1, 2$ *sont positifs et ceux commençant par* $3, 4, 5$ *sont négatifs. Ainsi,* 012344 *représente un nombre positif, alors que* 554342 *représente un nombre négatif.*

1.10 Représentation des entiers sur ordinateur

Dans un ordinateur, les nombres ne sont pas représentés en binaire "naturel". En effet, il est nécessaire de définir une taille de stockage fixe. Dans ce qui suit, nous supposerons que cette taille est de 1 octet, soit 8 bits. Dans les ordinateurs actuels, cette taille est généralement comprise entre 8 et 64 bits selon le type de variable employé. Par exemple, avec un ordinateur à base de processeur Intel Pentium tournant sous Linux, le type "short int" du langage C est stocké sur 16 bits, les types "int" et "long" sur 32 bits.

1.10.1 Le code binaire direct

On représente les entiers de la façon suivante.

- Un nombre entier positif ou nul est représenté par son expression binaire.
- Un nombre entier négatif est représenté par le complément à 2 de sa valeur absolue.

Autrement dit,
si le premier bit du nombre est 0, le nombre est positif ou nul,
si le premier bit du nombre est 1, le nombre est négatif.

***Exemple* 1.50.**
Le nombre 114 *est* $(1110010)_2$ *en base* 2, *et sa représentation sur un octet est* 01110010. *La représentation de* -114 *est donnée par le complément à* 2 *de* 114 *sur* 8 *bits, qui est* 10001110.

Regardons maintenant quels sont les entiers pouvant être représentés selon la méthode ci-dessus. Les entiers positifs ou nuls vont de $(00000000)_2$ à $(01111111)_2 = 127$. Les entiers négatifs vont de $(10000000)_2$ à $(11111111)_2$. Or $(10000000)_2$ est le complément à 2 de $(10000000)_2 = 128$, donc représente -128, et $(11111111)_2$ est le complément à 2 de $(00000001)_2 = 1$, donc représente -1. Ainsi, les nombres représentables sur 1 octet selon le code binaire direct sont:
$-128, -127, \ldots, -1, 0, 1, \ldots, 127$.

1.11 Représentation des nombres réels sur ordinateur

Un ordinateur ne manipule pas les nombres de la même façon qu'un être humain. En particulier, les nombres ne peuvent pas être de taille arbitraire, il est nécessaire de leur réserver une place de **taille fixe** pour la représentation sur ordinateur. Cette taille est déterminée par le programme en choisissant le type de la variable considérée parmi les types disponibles du langage de programmation utilisé.

Ainsi, un nombre entier est représenté par une variable de type entier et le code binaire direct vu précédemment. Les "grands" nombres entiers (qui excèdent la capacité de représentation du code binaire direct) ainsi que les nombres non entiers sont représentés par des variables de type réel et l'écriture exponentielle binaire.

1.11.1 L'écriture exponentielle binaire

Cette écriture est une normalisation (ou "standardisation"), c'est pourquoi on l'appelle aussi parfois écriture exponentielle normalisée. Tout nombre réel x non nul peut être représenté de la façon suivante,

$$x = (\text{signe})(\text{mantisse})_2 \cdot 2^{\text{exposant}} \tag{1.4}$$

avec

$$\text{signe} = \begin{cases} + & \text{si } x > 0 \\ - & \text{si } x < 0 \end{cases}$$

$$(0.1)_2 \leqslant (\text{mantisse})_2 < (1)_2$$

et "exposant" est un nombre entier donné en base 10. On remarquera en particulier la contrainte sur la mantisse qui doit être comprise entre $(0.1)_2 = 0.5$ et 1 (non inclus).

***Remarque* 1.51.** Bien que l'écriture donnée en (1.4) s'appelle écriture exponentielle binaire, l'exposant est donné en base 10.

***Exemple* 1.52.**
Utilisation de la représentation (1.4) pour transformer les nombres ci-après:

$$101.011 = 0.101011 \cdot 2^3$$
$$0.010101 = 0.10101 \cdot 2^{-1}$$
$$0.111 = 0.111 \cdot 2^0$$

Le nombre de bits réservés à la représentation sur ordinateur est fixe et est divisé en trois parties (voir figure 1.1). La première est constituée d'un seul bit pour le signe, la deuxième est pour l'exposant, et la troisième est pour la mantisse.

Supposons que l'on stocke un nombre réel sur 32 bits. On utilise alors le premier bit pour le signe s, puis, par exemple, les 7 bits suivants pour l'exposant, et les 24 bits restants pour la mantisse. Le signe s prend la valeur 0 si le nombre est positif ou nul, et 1 s'il est négatif. L'exposant dans l'écriture exponentielle binaire est un entier signé, mais n'est pas stocké en

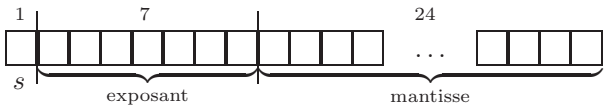

FIG. 1.1 – Stockage d'un nombre réel sur 32 bits.

utilisant un complément. On utilise la représentation dite par excès qui consiste à rendre positive la représentation des nombres en leur additionnant 2^{N-1} où N est le nombre de bits réservés pour le stockage, et en stockant la valeur ainsi trouvée appelée la caractéristique. L'entier 2^{N-1} est appelé le bias, c'est pourquoi cette représentation porte aussi le nom de représentation biaisée. Ainsi, si $N = 7$ (stockage de l'exposant sur 7 bits) on a la correspondance suivante:

$$\begin{aligned} \text{caractéristique} &= \text{exposant} + 2^{7-1} \\ &= \text{exposant} + 64\,. \end{aligned}$$

Finalement la mantisse est simplement stockée dans l'espace qui lui est réservé en ne stockant que les chiffres à la droite du point décimal.

***Remarque* 1.53.** En raison de la représentation (1.4) et des conditions qui lui sont imposées, on constate que la mantisse est un nombre fractionnaire dont la représentation en base 2 commence toujours par 0.1 (voir l'exemple 1.52). Dans la pratique, on ne stocke donc que les chiffres qui suivent ce 0.1 de tête, ce qui fait gagner un bit. Ce bit est appelé le bit caché. Cependant, dans la suite de notre exposé nous n'adopterons pas cette convention et la mantisse sera toujours stockée dans son intégralité.

***Remarque* 1.54.** La représentation du nombre zéro est un cas particulier où tous les bits sont nuls.

***Remarque* 1.55.** Une mantisse de 24 bits donne une précision d'environ 8 chiffres significatifs en base 10. Une mantisse de 53 bits donne une précision d'environ 16 chiffres significatifs en base 10.

***Remarque* 1.56.** Le standard IEEE 754 utilisé dans la plupart des processeurs actuels définit deux formats de nombres pour la représentation des réels, mais avec une normalisation différente de celle que nous avons présentée. La simple précision (32 bits) utilise un bit de signe, 8 pour l'exposant et 23 pour la mantisse, alors que la double précision (64 bits) utilise un bit de signe, 11 pour l'exposant et 52 pour la mantisse. Ce standard définit également des représentations spéciales pour l'infini et pour "NaN" ("Not a Number", résultat obtenu en cas d'opérations mathématiques illicites, comme par exemple la racine carrée d'un nombre négatif).

***Exemple* 1.57.**
Soit le nombre réel $x = 3.125$. *On convertit* x *en base* 2 *selon la procédure donnée à la section 1.6 pour obtenir* $(11.001)_2$. *L'écriture exponentielle binaire de ce nombre est donc* $(11.001)_2 = (0.11001)_2 \cdot 2^2$, *et la caractéristique est* $2 + 64 = 66 = (1000010)_2$. *La représentation de ce nombre sur* $1 + 7 + 24 = 32$ *bits est donc*

0	1000010	11001000000000000000000

***Exemple* 1.58.**
Soit le nombre réel $x = -0.15$. *La conversion en base* 2 *donne* $x = -(0.00\overline{1001})_2$, *et son écriture exponentielle binaire est* $x = -(0.\overline{1001})_2 \cdot 2^{-2}$. *La caractéristique est* $-2+64 = 62 = (111110)_2$. *Ainsi, la représentation de* x *est:*

1	0111110	10011001100110011001101