

Les amplitudes A peuvent être reconstituées aisément, pour les 512 valeurs par l'expression APL-ISO8485 suivante (en origine 0) :

$A \leftarrow + \setminus^{-1} 1 [S]$ Ⓐ Attention, en APL2, il faut parenthéser $^{-1}$
1

La figure sera obtenue par adressage sur l'écran graphique, d'une valeur sur deux seulement, constituant l'ordonnée de chacun des points représentatifs du graphe, le pas élémentaire valant 1 en abscisse, de sorte que l'on verra la forme de la sinusoïde dont l'amplitude variera entre $^{-8}1$ et 82 , représentée sur 256 points aussi bien avec $A[\square IO + 2 \times \imath 256]$ qu'avec $A[\square IO + ^{-1} + 2 \times \imath 256]$; On a en effet :

$256 \div 81 \quad 82$
3.160493827 3.12195122 Ⓐ Valeurs (suffisamment) proches
de π

Il est possible d'obtenir un tel codage de la sinusoïde pour toute valeur, imposée à l'avance, de la résolution horizontale ou verticale. Si l'écran est en haute définition (1024 pixels de large, donc horizontalement), le codage binaire équivalent de la même courbe aura 2048 bits de long (256 octets seulement).

On ne fera guère la différence entre le graphe obtenu avec ce codage binaire et celui que l'on calcule avec 1024 appels de la fonction sinus dont le vecteur de résultats, en double précision sous *APL* (64 bits par valeur), est alors codé en 8 kilo-octets. Quelle que soit la "précision" du calcul, tout graphe "courbe" devient une suite de segments qui deviennent eux-mêmes des séquences de pixels allumés chacun à une altitude discrète dépendant de la résolution (souvent plus de celle de la carte graphique, si la résolution de l'écran est meilleure que celle de la carte).

Le présent exposé cherche à transmettre quelques idées simplificatrices :

a) L'optimalité du codage de l'information d'un système physique ou d'un problème quelconque est atteinte *si et seulement si* tout bit du codage retenu est à la fois *nécessaire* et *suffisant* (pas de redondances, pas d'imprécision).

b) Ainsi codé, tout graphe ressort d'une algèbre parfaitement linéaire, celle de l'ensemble $\mathbf{Z}/2\mathbf{Z}$ (et des corps de Galois p -adiques), laquelle est aussi, directement, l'algèbre de l'ordinateur, car ses deux seuls nombres sont 0 et 1. Ces nombres ont la particularité, extraordinaire parmi tous les nombres possibles, d'être idempotents : Pour tout p positif, 0^p vaut 0 et 1^p vaut 1.

Une amplitude ne se distingue alors plus d'une intensité, ce qui peut simplifier considérablement le traitement du signal, lorsqu'on sait tirer parti de cette propriété. Monsieur *Fourier* ayant démontré, depuis fort longtemps, que tout signal pouvait se décomposer en sommes de sinusoides, il va de soi que la preuve exposée ici suffit à démontrer la généralité du concept : en effet, dans $\mathbb{Z}/2\mathbb{Z}$, toutes les grandeurs (les variables d'un système) seront exprimées, nécessairement soit par 0 ou par 1. Et ces grandeurs vont devenir strictement additives (ou soustractives), ce qui, **modulo 2**, s'exprime par une seule et même opération : \neg en APL, Ou Exclusif en général, (et XOR en jargon anglais).

c) Toutes les transformations que l'on avait l'habitude de considérer en algèbre traditionnelle, surtout avant l'invention, et l'utilisation, maintenant massive, des ordinateurs, peuvent aussi s'effectuer, directement, dans la mémoire de l'ordinateur, laquelle ne contient, c'est bien connu, que de longues suites de 0 et 1 et **JAMAIS** rien d'autre.

En résumé, on peut aisément se passer des codages de type logarithmique en base 2 que nous imposent, depuis un demi-siècle, les processeurs de machines et les types de données de tous les langages de programmation, *APL* inclus. Il suffit d'apprendre à se servir correctement de l'algèbre entière modulo 2 (malheureusement enseignée nulle part), et de tirer parti de ses propriétés, pour le moins extraordinaires, que la quasi-totalité des physiciens, et même des mathématiciens et des informaticiens, ignorent. *APL* permet, heureusement, d'expérimenter, de développer, d'enseigner cette algèbre merveilleuse et inconnue, isomorphe de l'algèbre logique, car *APL* est le seul langage de programmation (avec *J*, son fils) dans lequel 0 et 1 sont considérés, au choix mais automatiquement selon la fonction à appliquer, soit comme des états logiques opposés, soit comme des nombres entiers, et ce pour des structures de données quelconques, vecteurs, matrices (et hypertextes généralisés en *APL* étendu).

Depuis 1989, le *Laboratoire d'Informatique Théorique* (CEA, Saclay) a essayé de défricher les propriétés de l'algèbre de $\mathbb{Z}/2\mathbb{Z}$, pour reconstruire d'une manière simple et cohérente, l'ensemble des méthodes mathématiques applicables à la physique, sans considérer autre chose que des transformations (exprimées par des algorithmes optimaux) ne mettant en jeu que les grandeurs 0 et 1, avec lesquelles on peut exprimer tout signal et toute variation de signal, sans jamais utiliser d'autres nombres, sinon pour se raccorder aux méthodes enseignées résultant d'habitudes acquises, quant à la représentation des graphes variationnels par exemple, comme $y=f(x)$ ou $z=f(x,y)$.

Des représentations, classiques en mécanique, telle que celle de l'espace des phases, ont aussi leur équivalent dans $Z/2Z$. Il suffit d'aller les chercher, de les étudier, de les simplifier (sans axiomes contraires à l'observation) pour comprendre et modéliser bien des phénomènes, qui, autrement, vont demeurer incompris. En effet, lesdits phénomènes, exprimés par exemple à l'aide des équations traditionnelles de la physique, vont aboutir, en algèbre classique, à des formules *non linéaires*, approximatives en général; la plupart du temps, les systèmes d'équations différentielles qui résultent de la mise en équations de "réalités observables", ne sont pas intégrables; pour qu'ils deviennent intégrables, les physiciens émettent des hypothèses simplificatrices (bien souvent injustifiables physiquement), le seul but recherché étant alors de nature mathématique : permettre l'intégration, dans des cas très particuliers seulement, hélas, hélas, hélas...

Depuis que l'omniprésence des ordinateurs se fait sentir, on a, toutefois, remplacé la résolution analytique des équations compliquées, par une résolution numérique (souvent conduite, malheureusement, sans aucun moyen de vérification des résultats obtenus, sinon par d'autres calculs effectués à l'aide du même ordinateur ou de la même arithmétique flottante, tronquant l'information, comme chacun sait... et est bien obligé de s'en accommoder : faute de grives...).

Le choix délibéré de mettre au point de nouvelles méthodes utilisant avantageusement le langage de l'ordinateur et ses structures de données propres (mémoire linéaire contenant 0 ou 1 et rien d'autre) avait, *ab initio*, l'avantage certain que l'on allait *enfin* calculer juste, c'est-à-dire avec toute la précision requise dans chaque cas, sans jamais introduire de *chaos informatique* - que certains physiciens ont tendance, malheureusement, à confondre avec le chaos tout court, en tirant des conclusions dont il est aisé de prouver qu'elles sont erronées car elles résultent simplement de la *propagation* incontournable, si l'on persiste à vouloir traiter l'information avec des nombres soi-disant "réels" mais qui n'en sont jamais, des troncatures ("bit crunching") de l'arithmétique flottante...

Il est inutile de se servir de lunettes astronomiques, puis de télescopes de plus en plus puissants pour observer les galaxies lointaines : il existe une limite absolue, un flou imposé par la turbulence atmosphérique; la solution a, toutefois, récemment été trouvée : l'élimination de cette turbulence, par construction de télescopes fonctionnant dans des engins spatiaux, hors du flottement...

En ce qui concerne le traitement de données, la question se posait, rigoureusement de la même manière : seule l'élimination de la cause première des erreurs de calcul introduites non pas par les ordinateurs, mais par l'arithmétique avec laquelle on les fait fonctionner depuis 50 ans, permettait que l'on puisse espérer résoudre des problèmes de taille de plus en plus géante au fil des ans. Seule la reconsidération de ces problèmes en algèbre de $Z/2Z$ va permettre, dans tous les cas, de parvenir raisonnablement à ces fins. Le bit, 0 ou 1, est le *quantum d'information* absolu.

Il est *luxueux* de travailler avec une précision de calcul supérieure à celle des mesures; il est *dangereux* de travailler avec une précision de calcul inférieure à celle des mesures : les deux précisions peuvent être, pour que l'on atteigne l'optimum, du même ordre de grandeur, si l'on sait d'abord éliminer complètement les erreurs de calcul.

Comme on l'a montré sur la sinusoïde, et, par voie de conséquence, sur tout signal, 1024 valeurs de sinus en double précision, codées sur 8 kilo-octets ne contiennent pas plus d'information utile que la séquence minimale de bits 0 ou 1, qu'il est nécessaire d'envoyer sur une ligne de transmission, pour piloter un traceur graphique incrémental capable de reconstituer le graphe avec toute la précision nécessaire, pour que l'observateur humain soit *satisfait*, c'est-à-dire *incapable de faire la différence* avec un relevé expérimental dont le graphe serait exactement le même, compte tenu des fourchettes d'erreur de mesure tout au long de ce relevé, et de la résolution propre du traceur, bien entendu.

Par exemple, si l'erreur expérimentale atteint 1%, on n'observera pas mieux le phénomène en représentant son graphe sur 1024 points qu'en le représentant sur 100 points (disons 128 pour prendre la puissance de 2 supérieure la plus proche, car 64 pourrait dégrader l'information).

Alors, si le phénomène est perçu macroscopiquement comme sinusoïdal (il pourrait très bien ne pas l'être si l'on était capable d'effectuer l'observation, donc la mesure, avec une précision bien supérieure), il suffit de prendre un échantillonnage d'un bit sur deux (soit les bits pairs, soit les bits impairs) dans le code vectoriel binaire à 512 bits présenté plus haut, donc de ne conserver que 256 bits d'information soit 32 octets, pour obtenir une sinusoïde non différente de celle mesurable avec une précision deux fois moindre.

En outre, tout graphe non unidirectionnel, par exemple un dessin industriel (pour une maquette dessinée à l'échelle adéquate), peut être codé, rigoureusement, avec le même système, linéaire et optimal (car minimal), de manière paramétrique :

Une ligne de 0 et de 1 commande le déplacement élémentaire d'une plume de traceur dans une direction, soit vers le haut (1) soit vers le bas (0), d'un quantum égal à la moitié du *pas* (constante physique) du traceur selon cette direction; ce quantum peut s'appeler un "zig". (Avec un tel système, un code tel que 0 1 répété ne provoque pas de déplacement vertical de la plume : le trait obtenu est alors rigoureusement horizontal, car le micro-tremblement d'un demi-pas passe inaperçu : il faut deux zigs pour vaincre l'inertie).

Une seconde ligne de 0 et de 1 commande le déplacement élémentaire de la même plume, cette fois dans la direction perpendiculaire, soit vers la droite (1) soit vers la gauche (0), d'un quantum égal à la moitié du pas (toujours une constante physique) du traceur selon cette direction orthogonale à la précédente. (Les deux quanta peuvent être différents pour un traceur, ou un écran à balayage électronique, si ces appareils sont non *isotropes*). Appelons ce quantum un *zag*.

Une troisième ligne de 0 et de 1 commande le lever ou le baisser de la plume, donc l'absence ou la présence d'un trait sur le papier (quantum *zug*, d'origine germanique, "Zug" signifiant "trait" aussi bien que "train").

Il est aisé, avec le même codage paramétrique minimal, d'imaginer un traceur spatial (faisceau *laser*, allumé ou éteint) avec trois ou quatre lignes parallèles de 0 et de 1. Si le déplacement dans l'espace correspond à une *trajectoire* (que l'on peut appeler aussi bien *forme* qu'*attracteur étrange*), alors, celle-ci apparaît "continue"... par illusion due à nos sens, mais reste quantifiée : la quatrième ligne de 0 et 1 (éteint/allumé) est inutile quand le tracé correspond à un graphe bouclé.

Le présent raisonnement montre qu'une suite sur trois lignes parallèles d'informations binaires constitue un codage paramétrique optimal pour toute *forme* ou *information* de type tridimensionnel, ou pour toute forme bidimensionnelle avec solutions de continuité, comme dans l'écriture humaine ou dans un dessin au trait composé de plusieurs segments, tel qu'une *caricature*.

Aussi n'est-il pas étonnant que certaines formes de codage, très efficaces, de l'information, respectent déjà, et depuis longtemps, ce codage optimal.

Parmi les inventions humaines, on peut citer le code *Braille*, effectivement exprimé par des suites de parités 0 et 1 disposées sur deux ou trois lignes et balayées exactement comme préconisé ici (par les extrémités tactiles des doigts jouant un rôle de "scanner"). Des recherches ont même abouti à une normalisation internationale des dimensionnements et espacements.

Parmi les inventions non humaines, on peut citer le code génétique, effectivement exprimé par des suites de parités 0 et 1 également disposées sur deux ou trois lignes; ce code est encore plus optimisé, dans sa simplicité, que le code Braille (qui, par bien des points lui ressemble, ce qui prouve que l'on invente peu... en dehors de la reproduction, malgré soi - c'est-à-dire inconsciemment - des règles de *son propre code*...invisible, et inconnu de Louis Braille en 1826).

En effet, il n'existe que quatre lettres (U,A, C,G dans l'*ARN*) et (T,A, C,G dans l'*ADN*). Ces lettres sont les symétriques l'une de l'autre deux à deux, de sorte qu'il n'est besoin, en réalité, que de deux lettres et de leurs symétriques, à assembler en mots de trois lettres (un peu comme si notre alphabet ne comportait que les quatre lettres "b" et "d" d'une part, "p" et "q" d'autre part, et que l'on ne puisse former, avec ces consonnes, que des radicaux de trois lettres sans voyelles, comme c'est d'ailleurs le cas pour la formation des radicaux dans les langues sémitiques). Il est d'ailleurs très facile de "voir" le codage binaire correspondant dans les *formules chimiques* des bases (par exemple, dans l'*ARN*, l'Uracile-Adénine, miroir de l'Adénine-Uracile, et la Cytosine-Guanine, miroir de la Guanine-Cytosine) en codant 1 l'**azote** (N), et 0 l'**oxygène** (O) porteurs des liaisons hydrogène (symbolisées par ":") de deux types, soit O:H-N soit N-H:N ; c'est de l'*information chimique*, pure et simple :

T-A correspond alors à	0 1	et, par retournement,	A-T à	1 0
(U-A dans l' <i>ARN</i>)	1 1	(A-U dans l' <i>ARN</i>)		1 1
C-G correspond alors à	1 0	et, par retournement,	G-C à	0 1
	1 1			1 1
	0 1			1 0

Or, de telles matrices (qui, effectivement, sont les matrices de toutes les programmes possibles de tous les êtres vivants) *ne* sont *pas* quelconques.

Elles ont des propriétés extraordinaires si on les étudie en algèbre des parités (celle de **Z/2Z**) et non pas en tenant compte de potentiels de charges (l'habitude du calcul macroscopique...). Car le code de l'*ADN* est de l'*information* pure, à la fois *programme* et *données*. Et il n'est pas écrit en langage LISP : il est matriciel.

0 1	est, modulo 2, racine cubique de la matrice-unité,	1 1
1 1	et inverse de son symétrique par rapport au centre :	1 0

(le centre est un point virtuel au centre du carré de la matrice 2 2).

1 0	est, modulo 2, racine carrée de la matrice-unité,	1 1
1 1	et constitue, avec sa transposée ci-contre :	0 1

un couple de matrices involutives non triviales (donc capables de produire des transformées d'un signal discret à deux bits) permettant de relier l'algèbre génétique à Fourier, Laplace, Hadamard, Walsh en théorie du signal.

Comme on peut le montrer, certaines propriétés du code génétique correspondent dans $\mathbb{Z}/2\mathbb{Z}$, à ce que l'on va observer, par double produit matriciel (voir, par ailleurs les simplifications algorithmiques de la transformation de Hadamard et la théorie de la vision expliquée par la transformation cognitive - APL94), algébriquement, à condition que l'on considère ces codes effectivement à la fois comme des programmes et des données.

Les habitués de la notation d'Iverson pourront aisément vérifier que dans la paire A-T binarisée, chaque ligne est donnée par $\neq \setminus$ de l'autre.

Et, si on attache 1 devant les lignes de la paire CG, on vérifie les relations :

$\neq \setminus$ 1 1 1
1 0 1

$\neq \setminus$ 1 0 1
1 1 0

Celles-ci permettent de déduire (en ôtant ensuite le 1 rajouté à gauche) les lignes extrêmes à partir de la ligne médiane. Et la matrice de rang 3, avec cette colonne de 1 telle que le déterminant ne puisse être nul, est alors matriciellement racine septième de la matrice-unité, modulo 2. Aucune de ces propriétés n'apparaît en algèbre normale, si l'on considère des nombres autres que 0 et 1.

En outre, les <i>puissances matricielles successives</i> de la matrice	1 1
isomorphe de la <i>matrice chromosomique</i> de <u>différenciation sexuelle</u>	1 0
	XX
	XY

(X, chromosome plus grand que Y, étant symbolisé par 1, et Y par 0), révèlent le caractère fibonnaccien de la reproduction sexuée (comme le trouva Fibonacci à la fin du XIIIe siècle en étudiant... la reproduction des lapins et des lapines), chaque matrice étant la somme des deux précédentes et ne contenant, dans ses lignes et ses colonnes, QUE des termes successifs de la suite de Fibonacci.

L'étude plus poussée de l'algèbre **matricielle** de $\mathbb{Z}/2\mathbb{Z}$ permettra-t-elle de découvrir des secrets mystérieux des codages de l'information, (notamment en biologie), totalelement inaccessibles en algèbre classique ? C'est probable, à condition que physiciens, généticiens et biologistes moléculaires se mettent à sonder cette algèbre - passionnante, (car la symétrie y règne, rien n'y est **ni aléatoire ni statistique**) - de très près. Et l'apprentissage nécessaire est beaucoup moins difficile que celui de la théorie des fonctions..., fruit de notre culture mathématique, mais qui s'avère inadaptée à l'étude correcte de **l'information**, par essence *discontinue*, (comme la chimie qui la gère en nous de A à Z, de α à ω).

Remarques

La fonction SIN512 n'est pas un convertisseur de sinusoïde en bits. Le lecteur peut, à titre d'exercice, écrire aisément cette fonction inverse de $A \leftrightarrow S^{-1}$ qui transforme donc A, l'amplitude, en S, signal binaire linéaire, également en fonction de la résolution verticale choisie. La pente maximale de la sinusoïde vaut 1 en valeur absolue, si on considère le graphe de $\sin(x)$ avec x en radians entre 0 et 2π . Si $2N$ est le nombre vertical de pixels correspondant à l'intervalle de variation 2π , entre -1 et $+1$, il faudra $2\pi N$ bits (voir la formule d'incertitude de Heisenberg...) pour avoir la certitude que la fonction *sinus* soit représentée au mieux et de manière isotrope sur les deux axes sur un écran lui-même isotrope. Les 512 bits de l'exemple sont tout à fait suffisants pour l'affichage d'une sinusoïde "harmonieuse", dans le but d'illustrer un cours de mathématiques ou de physique. (Un signal de pente maximale élevée sera, au contraire, beaucoup mieux codé par une double suite de bits, donc en *paramétrique*).

Bien entendu, il est possible de comprimer encore le contenu des 512 bits de S, mais cela ne vaut pas la peine, car tout fichier ou variable a, souvent, avec son en-tête, une taille minimale supérieure à ces 64 octets du seul contenu.

De plus, le but de cet article est surtout de faire ressortir l'intérêt de considérer préférentiellement toute expression de l'information comme du strict binaire en remplaçant - avantageusement - toute numérisation par des codes linéaires immédiatement utilisables, surtout si le codage devient paramétrique, dans une foule d'applications, du pilotage de traceurs à la scintigraphie; (les codes paramétrés ressemblent à du braille, que certains non non-voyants exercés lisent aussi vite qu'ils lisent un texte imprimé avec notre alphabet latin; on peut même ajouter qu'à force de voir du braille, accompagné de sa transcription littérale, on finit par savoir le lire même sans l'apprendre; tel est le cas des bibliothécaires d'Instituts spécialisés comme à Paris, celui des Jeunes Aveugles. Ce code a beaucoup à nous apprendre quant aux mécanismes qui régissent la perception (substitut de la vision) et la cognition, car il est proche de l'optimum du code génétique et se lit par propagation (synonyme de cumul ou de balayage - en anglais "scan" - de *différences*, en APL $\neq \setminus$ comme on aurait pu s'en douter..., sauf si l'on ignore APL et son idiome *il-de-lynx*).