

Jeux APL
par Gérard Langlet

La Rétroversion de l'Inversion Diabolique (ou : l'Essence sans PB des Pompes à Shadoks)

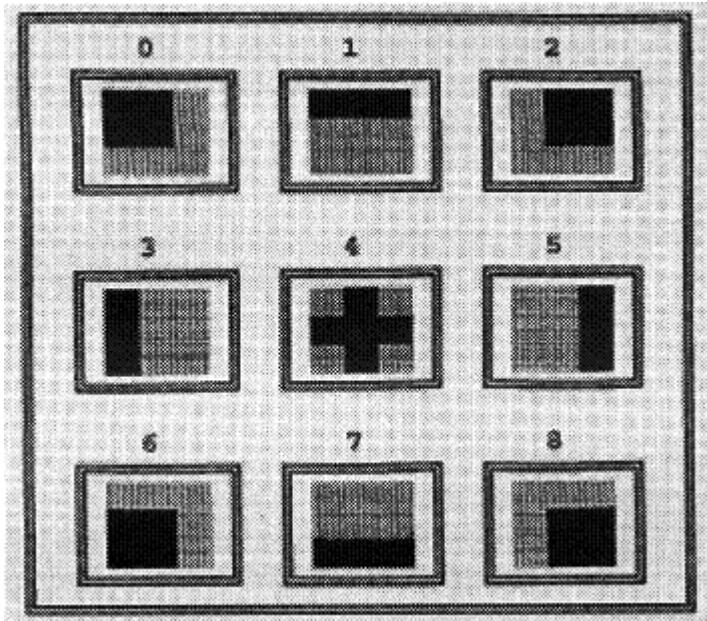
Résumé

Cet article est le premier d'une série consacrée à la Théorie des Jeux, revue et amplement corrigée, grâce à l'exposé progressif des propriétés de \neq \ l'intégré-différenciation binaire ou propagation asymétrique de la parité ou encore interaction élémentaire en physique.

Il va se baser sur l' "Inversion Diabolique" [1], jeu également appelé "Magic" par les Anglais, friands de puzzles, voir [2]; ce jeu consiste à tirer au hasard une configuration C initiale dans un carré 3x3, de sous-carrés noirs OU blancs, puis à choisir, parmi une palette de 9 transformations possibles, numérotées de 0 à 8, la succession minimale des transformations qui vont, petit à petit, "inverser" partiellement les couleurs des sous-carrés de la configuration C jusqu'à atteindre une autre configuration B appelée le but, imposée elle aussi (tirée en général également au hasard); parfois, des logiciels sophistiqués permettent de se battre contre l'ordinateur qui gagne s'il est bien programmé; il peut même apprendre, comme mentionné en [1], la stratégie de l'adversaire, et se servir de ses acquis pour mieux battre les candidats suivants.

Nous allons montrer comment ce jeu se désosse complètement grâce à APL, et faire entrevoir que la résolution de ce puzzle n'est qu'une première étape vers la résolution générale, par la même voie, de TOUS les puzzles imaginables.

Introduction et Rappels

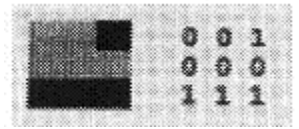


Voici les 9 transformations admises au menu (voir [1]) :

Chacune d'entre elles inverse la couleur de trois (1, 3, 5 et 7) carrés sur les 9, de quatre carrés (0, 2, 6 et 8), ou, comme la croix centrale (4), de cinq. On doit être capable, à partir d'une configuration C quelconque, de cliquer sur une des 9 icônes et de jouer, pour parvenir au but B en un nombre minimal de ces inversions diaboliques, de préférence avant l'ordinateur

M_0	M_1	M_2
1 1 0	1 1 1	0 1 1
1 1 0	0 0 0	0 1 1
0 0 0	0 0 0	0 0 0
M_3	M_4	M_5
1 0 0	0 1 0	0 0 1
0 0 0	1 1 1	0 0 1
0 0 0	0 1 0	0 0 1
M_6	M_7	M_8
0 0 0	0 0 0	0 0 0
1 1 0	0 0 0	0 1 1
1 1 0	1 1 1	0 1 1

Si la matrice C est la configuration telle que la propose J.J. Girardot dans la réf. [1] à la page 55, l'application de l'UNE des 9 matrices M_n (à gauche) par un simple OU EXCLUSIF scalaire produit nécessairement la configuration optimale suivante. Autrement dit, si M est un tableau de dimension 9 3 3 2, contenant toujours dans sa première hyper colonne, le tableau M de dimension 9 3 3 de ces 9 matrices, et, dans sa seconde hypercolonne le tableau 9 3 3 contenant C répété 9 fois, les 9 configurations suivantes possibles, dont une est la bonne, sont données par la seconde hypercolonne de $\neq \setminus M$ ou par \neq / M .



Un damier 3x3 donc à 9 cases $\in \{0, 1\}$ forcerait à considérer $2 * 9$ soit 512 combinaisons possibles, (réductibles à un nombre bien plus faible si l'on tient compte des symétries du carré). Nous verrons, in fine, que c'est inutile.

Toutes les configurations possibles sont représentables par décomposition d'indices en octal (à ne pas confondre avec l'indice d'octane, voir plus loin), puis en binaire, des nombres de 0 à 511. 9 décompositions correspondent à M:

$M \leftarrow \text{2 2 2} \text{ 8 8} \text{ 4 3 2 4 4 8 2 1 5 2 9 2 1 8 5 7 3 5 4 7 2 7}$

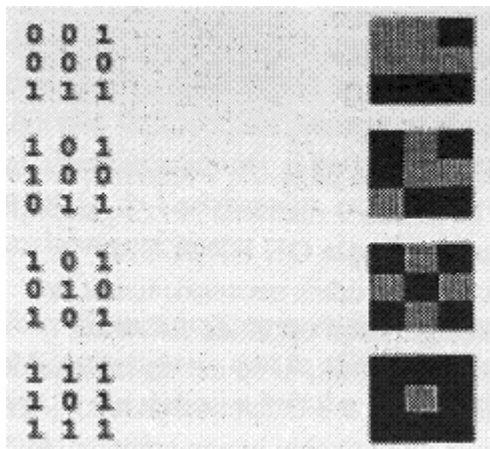
Avec la configuration C listée plus haut, on vérifie que les matrices M_3 , M_6 et M_4 également dans les plans 3 6 et 4 de M (en origine 0) par actions successives sur C, d'ailleurs dans n'importe quel ordre, car \neq est aussi PLUS modulo 2, donc une fonction commutative, fournissent bien le but à atteindre:

$\neq / M [3 \ 6 \ 4 ; ;]$



[3] par Bernard Mailhol, et la liste [4].

$\neq \setminus [0] C ; M [3 \ 6 \ 4 ; ;]$



Les étapes successives sont décrites, de C à la configuration du but à atteindre, par la Propagation symétrique de la Parité, autrement dit $\neq \setminus$ l'interaction élémentaire universelle, exposée dans divers articles dont

$\neq \setminus$ EST UN IDIOME INTELLIGENT:

Comme on le conjecture maintenant pour TOUS les jeux et puzzles possibles, $\neq \setminus$ serait CAPABLE de jouer et de gagner même aux échecs; autant dire que, pour lui, l'inversion diabolique reste un jeu de potaches, intéressant surtout à titre pédagogique pour montrer la puissance d'APL et même celle des extensions que ce langage n'a pas encore l'honneur de posséder... (sans rapport avec l'épithète qualifiant l'inversion).

Dans ce puzzle, l'application de deux fois la même inversion ne servirait à rien, car on reproduirait la configuration originale. Autrement dit, s'il existe neuf

matrices permises, et que l'inventeur du jeu a, un jour, vérifié ou prouvé qu'il existait toujours une solution, cela implique que l'on doit utiliser au maximum une seule fois chaque matrice mise en jeu. Mais nous allons le redémontrer.

B

1	1	1
1	0	1
1	1	1
		<i>+D←B≠C</i>
1	1	0
1	0	1
0	0	0

Si *B* est la matrice du but à atteindre et si *C* est celle de la configuration courante, la différence $D \leftarrow B \neq C$ est la somme modulo 2 des matrices à appliquer une fois et une seule au plus, parmi les 9 matrices. La solution correspondra donc à un simple vecteur de 9 coefficients *V* valant soit 0 soit 1, dont il est possible d'écrire le système d'équations paritaires c'est-à-dire en algèbre modulo 2 directement, donc en logique binaire pure.

On rappellera que \neq diadique symbolise à la fois l'exclusion logique (OU EXCLUSIF), plus modulo 2 et moins modulo 2.

Dans chaque position du damier, les bits sont indépendants; leur futur ne dépend pas des autres. S'il existe 9 bits, il faut effectivement 9 coefficients, pour avoir 9 équations; il faut et il suffit alors, pour que le puzzle soit toujours soluble, disposer de 9 matrices indépendantes, donc non déductibles les unes des autres par une application linéaire laquelle est nécessairement, en algèbre modulo 2, une simple somme, donc, en logique, une réduction par OU EXCLUSIF.

Soient *a b c d e f g h i* les 9 coefficients inconnus $\in \{0, 1\}$ composantes d'un vecteur *V* inconnu, le "Vecteur stratégique" ou masque-mystère contenant 1 si on doit appliquer l'inversion diabolique *Mn* OU 0 si on doit l'ignorer.

On devra avoir, TERME à TERME:

$$D \equiv (a \times M0) \neq (b \times M1) \neq (c \times M2) \neq \dots \neq (i \times M8) \wedge$$

ou, car, en binaire, le produit correspond à ET:

$$D \equiv (a \wedge M0) \neq (b \wedge M1) \neq (c \wedge M2) \neq \dots \neq (i \wedge M8)$$

On va former un système linéaire logique.

Soit *d o I* une fonction-boucle dont la syntaxe est: *e d o I n* et qui exécute *n* fois son argument gauche *e* contenant une expression APL, en faisant automatiquement varier un indice local *I* de 1 à *n* en origine 1 ou de 0 à *n-1* en origine 0 :

'*Y[I;]←,M[I;;]*' *d o I* 1↑ρ*Y*←9 9ρ0

Y

1	1	0	1	1	0	0	0	0
1	1	1	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0
1	0	0	1	0	0	1	0	0
0	1	0	1	1	1	0	1	0
0	0	1	0	0	1	0	0	1
0	0	0	1	1	0	1	1	0
0	0	0	0	0	0	1	1	1
0	0	0	0	1	1	0	1	1

DD

1	1	0	1	0	1	0	0	0
0	0	0	1	1	0	1	0	0
1	1	0	1	0	1	0	0	0

PB Y

Les lignes contiennent *M0, M1, ...*

Le VECTEUR *V* est tel que le produit matriciel modulo 2 de *V* par la matrice *Y* doit reproduire *DD* c'est à dire la différence $B \neq C$ linéarisée.

Le puzzle est solution d'un système linéaire en algèbre modulo 2 isomorphe de l'algèbre binaire. Une de nos conjectures actuelles consiste à dire qu'une telle proposition sera vraie pour tout puzzle en particulier et pour tout problème soluble y compris la théorie unitaire du champ en physique (d'Einstein), le bouclage du budget de l'Etat en économie et la prévision météorologique précise, à condition de renoncer définitivement à décrire les phénomènes, comme on l'a fait jusqu'ici depuis plusieurs siècles, par fonctions continues, des nombres autres que 0 et 1, et surtout, hélas, par des statistiques.

Dans la dernière expression ci-dessus, la fonction PB est l'essence même du "moteur du système" (en anglais BP pour "Binary Product" à indice d'octane élevé mais sans lourdeur donc sans plomb, [symbole chimique Pb], capable de résoudre TOUS les problèmes - en argot de potache: PB).

PB représente le Produit Binaire généralisé (scalaire et matriciel à la fois), défini en APL comme:

$$PB : \alpha \neq . \wedge \omega \quad \text{ou} \quad PB : 2 | \alpha + . \times \omega$$

si l'on adopte la notation $\alpha \ \omega$ chère à Ken Iverson.

Soit IBM (Inversion Binaire Matricielle, comme dirait l'Ingénieur Bernard Mailhol) la fonction APL capable de simuler le Domino Binaire (à normaliser d'urgence comme le souhaiterait peut-être Raymond Tisserand sous le nom de 2e DB, car APL a déjà depuis longtemps un premier domino dû à Mike Jenkins). On peut laisser au lecteur le soin de programmer cette fonction IBM, s'il n'a pas la patience de lire cet article jusqu'au bout...

Alors, nécessairement, lorsqu'une matrice est inversible, donc possède un déterminant non nul, en algèbre modulo 2 comme en algèbre "normale", le système a une unique solution, comme tout système linéaire classique, donnée par la formule magique de la potion d'Astérix.

Voici IY l'inverse matriciel classique de Y :

$$+IY \leftarrow \frac{5}{2} Y$$

-0.2	0.6	-0.2	0.6	-0.2	-0.4	-0.2	-0.4	0.8
0.4	-0.2	0.4	-0.2	0.4	-0.2	-0.6	0.8	-0.6
-0.2	0.6	-0.2	-0.4	-0.2	0.6	0.8	-0.4	-0.2
0.4	-0.2	-0.6	-0.2	0.4	0.8	0.4	-0.2	-0.6
0.4	-0.2	0.4	-0.2	-0.6	-0.2	0.4	-0.2	0.4
-0.6	-0.2	0.4	0.8	0.4	-0.2	-0.6	-0.2	0.4
-0.2	-0.4	0.8	0.6	-0.2	-0.4	-0.2	0.6	-0.2
-0.6	0.8	-0.6	-0.2	0.4	-0.2	0.4	-0.2	0.4
0.8	-0.4	-0.2	-0.4	-0.2	0.6	-0.2	0.6	-0.2

Le déterminant de la matrice Y vaut 5 et, en conséquence, le déterminant de IY vaut 0.2. Cette dernière inverse IY multipliée par 5 sera la matrice entière de gauche, dont la parité, reste de la division entière par 2, est à droite:

$5 \times IY$									$+IYB \leftarrow 2 5 \times IY$								
-1	3	-1	3	-1	-2	-1	-2	4	1	1	1	1	1	0	1	0	0
2	-1	2	-1	2	-1	-3	4	-3	0	1	0	1	0	1	1	0	1
-1	3	-1	-2	-1	3	4	-2	-1	1	1	1	0	1	1	0	0	1
2	-1	-3	-1	2	4	2	-1	-3	0	1	1	1	0	0	0	1	1
2	-1	2	-1	-3	-1	2	-1	2	0	1	0	1	1	1	0	1	0
-3	-1	2	4	2	-1	-3	-1	2	1	1	0	0	0	1	1	1	0
-1	-2	4	3	-1	-2	-1	3	-1	1	0	0	1	1	0	1	1	1
-3	4	-3	-1	2	-1	2	-1	2	1	0	1	1	0	1	0	1	0
4	-2	-1	-2	-1	3	-1	3	-1	0	0	1	0	1	1	1	1	1

On vérifiera que le produit matriciel de y par IYB à gauche, est une matrice entière dont la parité terme à terme est une jolie matrice-unité :

$Y+ . \times IYB$	$2 Y+ . \times IYB$
1 4 2 4 2 2 2 2 2	1 0 0 0 0 0 0 0 0
2 3 2 2 2 2 2 0 2	0 1 0 0 0 0 0 0 0
2 4 1 2 2 4 2 2 2	0 0 1 0 0 0 0 0 0
2 2 2 3 2 0 2 2 2	0 0 0 1 0 0 0 0 0
2 4 2 4 1 4 2 4 2	0 0 0 0 1 0 0 0 0
2 2 2 0 2 3 2 2 2	0 0 0 0 0 1 0 0 0
2 2 2 4 2 2 1 4 2	0 0 0 0 0 0 1 0 0
2 0 2 2 2 2 2 3 2	0 0 0 0 0 0 0 1 0
2 2 2 2 2 4 2 4 1	0 0 0 0 0 0 0 0 1

On vérifiera aussi que IYB est bien, en algèbre modulo 2, matrice inverse (à gauche et, bien sûr, à droite) de Y :

$Y \neq . \wedge IYB$	$IYB \neq . \wedge Y$
1 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0	0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0	0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0	0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0	0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0	0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0 1

V sera le vecteur des transformations à appliquer (dans n'importe quel ordre) à cc la configuration courante C linéarisée, pour obtenir le but linéarisé BB :

B								
1	1	1						
1	0	1						
1	1	1						
$+BB \leftarrow, B$								
1	1	1	1	0	1	1	1	1
$+CC \leftarrow, C$								
0	0	1	1	0	0	1	1	1
$+DD \leftarrow, C \neq B \quad \text{ou} \quad +DD \leftarrow CC \neq BB$								
1	1	0	0	0	1	0	0	0

```

V
0 0 0 1 1 0 1 0 0
a b c d e f g h i

```

```

V/19  A avec PP=3
3 4 6

```

```

+R←V +.× Y
1 1 0 3 2 1 2 2 0

```

```

R +.× IY  A avec PP=3
-7.22E-16 1.11E-16 5.83E-16 1
1 -5E-16 1 3.89E-16
3.33E-16

```

```

2|R +.× IY
0 0 0 1 1 0 1 0 0  A donc V

```

```

CT<| R +.× IY
0 0 0 1 1 0 1 0 0  A donc V

```

```

DD ≠.∧ IYB
0 0 0 1 1 0 1 0 0

```

On vérifiera aisément en effectuant le produit matriciel normal de V par Y que l'on obtiendra un résultat R tel que si on multiplie matriciellement ce résultat par IY la matrice inverse de Y, on obtient un vecteur, certes non entier, mais tel que ses termes non pollués par d'affreux résidus d'erreur d'arrondi hélas toujours provoqués par l'arithmétique flottante de papa, fournissent les indices cherchés. On corrige (cela varie en outre selon les interprètes APL utilisés !)

soit en prenant la parité, reste de la division entière par 2

soit en négligeant ces erreurs C.Q.F.D.

Mais il est plus efficace de trouver par le produit matriciel binaire directement. Re-C.Q.F.D.

On aura réussi à démontrer, d'une part, qu'il suffira de raisonner correctement sur le concept de parité SEUL pour résoudre les puzzles, et, d'autre part, qu'il manque bien un concept essentiel dans TOUS les langages de programmation, APL inclus, celui de l'inversion matricielle modulo 2 ou binaire, **PROVISOIREMENT** dans les limbes.

Le mot **PROVISOIREMENT** est ici mis en gras majuscule exprès: la suite va tenter de montrer pourquoi et comment on peut se passer de ce concept, encore une fois grâce au remue-méninges mathématique ("brainstorming") que seul permet APL, non seulement par son approche globale de la programmation, mais comme outil bon marché sinon gratuit, de recherche, de réflexion et de simplification de TOUT PROBLÈME avant de foncer tête baissée vers les outils de calcul, fruits de trois siècles de mathématique du nombre; la dualité algèbre binaire/algèbre modulo 2, sujet inexploré quant aux propriétés de symétrie et d'asymétrie de l'information considérée en vecteurs et en matrices, fortement liée aux propriétés mêmes de la matière, étudiées en physique des particules et en cristallographie, mais jusqu'ici ignorées en informatique, va jouer son rôle.

De la mathématique du nombre à celle des Parités

Elever une matrice binaire en puissances successives consiste à la multiplier matriciellement par elle-même modulo 2 plusieurs fois.

Voici alors le carré, le cube et le bi-carré (carré du carré) de Y:

$+Z \leftarrow Y$	PB	Y	Z	PB	y	Z	PB	Z
1	1	1	1	0	1	1	1	0
0	1	0	1	0	1	0	0	0
1	1	1	1	0	1	0	1	1
0	1	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	1	0
1	1	0	1	0	1	1	1	1
0	0	0	1	0	1	0	1	0
0	1	1	1	0	1	1	1	1
1	1	1	1	1	0	1	0	0
0	1	0	1	0	1	1	0	1
0	1	1	1	0	0	0	1	1
0	1	0	1	1	1	0	1	0
1	1	0	0	0	1	1	1	0
1	0	0	1	1	0	1	1	1
1	0	1	1	0	1	0	1	0
0	0	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1

Dans tous les cas, ou bien la matrice va produire, par itération du produit matriciel binaire (PB) avec elle-même, soit une matrice-unité à une certaine puissance n , soit autre chose; cet autre chose peut correspondre à une matrice nulle, dégénérée (cycler sur un motif périodique ne reproduisant jamais l'original: il y a alors cancérisation, alors que la $(n+ 1)$ ième puissance d'une matrice inversible est la matrice elle-même, saine).

Voici ensuite le carré, le cube et le bi-carré de IYB l'Inverse Binaire Matriciel de Y :

1	1	1	1	0	1	1	1	0
0	1	0	1	0	1	0	0	0
1	1	1	1	0	1	0	1	1
0	1	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	1	0
1	1	0	1	0	1	1	1	1
0	0	0	1	0	1	0	1	0
0	1	1	1	0	1	1	1	1
1	1	0	1	1	0	0	0	0
1	1	1	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0
1	0	0	1	0	0	1	0	0
0	1	0	1	1	1	0	1	0
0	0	1	0	0	1	0	0	1
0	0	0	1	1	0	1	1	0
0	0	0	1	0	1	0	1	1
0	0	0	0	0	0	1	1	1
0	0	0	0	1	1	0	1	1
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1

On en déduit que les DEUX matrices sont des racines quatrièmes de l'unité, en algèbre modulo 2.

En conséquence, et ceci est vrai pour toute matrice comme pour tout nombre: si le cube d'une matrice quelconque, multiplié matriciellement par ladite matrice, a pour résultat une matrice-unité, alors, nécessairement, ledit cube est aussi la matrice inverse de ladite matrice.

Ceci se vérifie VISUELLEMENT, en moins de temps qu'il ne faut pour écrire, en APL :

$$IYB \equiv 2 | Y + . \times Y + . \times Y \quad \text{A OU } IYB \equiv Y \neq . \wedge Y \neq . \wedge Y$$

$$1 \quad \text{A OU } Y \equiv IYB \neq . \wedge IYB \neq . \wedge IYB$$

En algèbre des parités, les théorèmes tombent en cascade :

Si une matrice quelconque est inversible, il est nécessaire qu'elle soit racine nième de la matrice-unité de même rang (au sens mathématique et non au sens APL); alors, la $(n-1)$ ième puissance de la matrice est nécessairement sa matrice inverse.

Le domino binaire se simplifie... conceptuellement, sans calcul. Au lieu de diviser, on multiplie... Plus de calcul de déterminant (toute matrice inversible a son déterminant égal à 1 en algèbre modulo 2, car ce dernier ne peut être nul, et aucune autre valeur que 0 OU 1 n'est permise).

Dans tous les cas, ou bien la matrice va produire, par itération du produit matriciel binaire (PB) avec elle-même, soit une matrice-unité à une certaine puissance n , soit autre chose; cet autre chose peut correspondre à une matrice nulle, dégénérée (cycler sur un motif périodique ne reproduisant

jamais l'original: il y a alors concréisation, alors que la (n+ 1)ième puissance d'une matrice inversible est la matrice elle-même, saine).

Toutes les matrices inversibles, si on les réordonne artistement, doivent posséder une permutation de lignes (ou, par symétrie, de colonnes) pour laquelle une diagonale ne contiendra que des 1 (tel est le cas de M); la condition est nécessaire, mais non suffisante (une matrice pleine de 1 n'est pas inversible) .

Dans le cas du puzzle dit "l'inversion diabolique", chaque matrice M effectue une opération qui, on l'a vu, si elle est pratiquée deux fois, restaure la configuration originale.

Ceci implique, si le puzzle est bien un puzzle soluble:

$Y \wedge . = \diamond Y$ \wedge idem pour IYB

```

1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 1

```

Connexion fractale

Voici un autre puzzle sur le puzzle et sa solution...

On veut fabriquer la matrice de droite G9 ci-après.

A gauche de chaque ligne se trouve l'expression APL qui a pour résultat la ligne en question.

On voit que chaque ligne est une combinaison linéaire des lignes de Y (identifiées par leur

numéro nl entre 0 et 8).

nl	Y	Expression E APL	Résultat
0	1 1 0 1 1 0 0 0 0	$\neq Y[0\ 2\ 4\ 6\ 8;]$	1 1 1 1 1 1 1 1 1
1	1 1 1 0 0 0 0 0 0	$\neq Y[1\ 3\ 5\ 7;]$	0 1 0 1 0 1 0 1 0
2	0 1 1 0 1 1 0 0 0	$\neq Y[0\ 1\ 2\ 5\ 6\ 8;]$	0 1 1 0 0 1 1 0 0
3	1 0 0 1 0 0 1 0 0	$\neq Y[0\ 3\ 7\ 8;]$	0 1 0 0 0 1 0 0 0
4	0 1 0 1 1 1 0 1 0	$\neq Y[0\ 3\ 5\ 6\ 8;]$	0 1 1 1 1 0 0 0 0
5	0 0 1 0 0 1 0 0 1	$\neq Y[2\ 5\ 6\ 7;]$	0 1 0 1 0 0 0 0 0
6	0 0 0 1 1 0 1 1 0	$\neq Y[0\ 2\ 3\ 4\ 6;]$	0 1 1 0 0 0 0 0 0
7	0 0 0 0 0 0 1 1 1	$\neq Y[1\ 3\ 5\ 6\ 8;]$	0 1 0 0 0 0 0 0 0
8	0 0 0 0 1 1 0 1 1	$\neq Y[0\ 1\ 2\ 3\ 4\ 6;]$	1 0 0 0 0 0 0 0 0

G9

À l'aide de la fonction $d \circ I$, on peut faire imprimer les lignes successives donnant l'expression et son résultat formaté correctement pour rédiger cet article:

```

'K←G9[I;]≠.∧IYB ∘ K←≠K/19 ∘ E,≠±E←20↑2φ'';]≠Y['',K' dOI 9
≠Y[0 2 4 6 8;]      1 1 1 1 1 1 1 1 1
≠Y[1 3 5 7;]        0 1 0 1 0 1 0 1 0
≠Y[0 1 2 5 6 8;]    0 1 1 0 0 1 1 0 0
≠Y[0 3 7 8;]        0 1 0 0 0 1 0 0 0

```



```

≠Y[0 3 5 6 8;]      0 1 1 1 1 0 0 0 0
≠Y[2 5 6 7;]        0 1 0 1 0 0 0 0 0
≠Y[0 2 3 4 6;]      0 1 1 0 0 0 0 0 0
≠Y[1 3 5 6 8;]      0 1 0 0 0 0 0 0 0
≠Y[0 1 2 3 4 6;]    1 0 0 0 0 0 0 0 0

```

Le Produit Binaire de chaque ligne de $G9$, que l'on veut obtenir comme s'il s'agissait d'une configuration finale du puzzle, par IYB , met dans K les numéros de transformations à appliquer pour y parvenir à partir d'une configuration blanche ou nulle; on vérifie en réexécutant l'expression E .

Mais, en plus simple, on peut avec un seul produit, obtenir TOUS les masques désirés, pour toutes les lignes de $G9$:

```

G9 ≠ . ^ IYB
1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0
0 1 1 0 0 1 1 0 1
1 0 0 1 0 0 0 1 1
1 0 0 1 0 1 1 0 1
0 0 1 0 0 1 1 1 0
1 0 1 1 1 0 1 0 0
0 1 0 1 0 1 1 0 1
1 1 1 1 1 0 1 0 0

```

En remplaçant $G9$ par T obtenu comme:
 $T \leftarrow \text{q}(9\rho 2) \top 1512$, on obtient les 512 masques de toutes les configurations possibles finales du puzzle. Pour passer de CC à la configuration BB (le but), il suffira d'appliquer ≠ BIT à BIT entre CC et la ligne de $T \neq . ^ IYB$ correspondant à la représentation binaire de BB dans T .

La même démarche de pensée s'appliquerait à des puzzles bien plus compliqués.

```

T[2 1 CC ≠ BB] ≠ . ^ Y ≠ . ^ Y ≠ . ^ Y
0 0 0 1 1 0 1 0 0

```

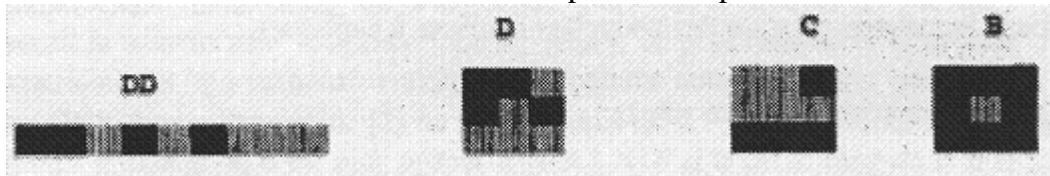
En simplifiant, pour toute matrice B de dimension 3 3 de configuration à atteindre appelée BUT, les numéros de transformation à partir d'une matrice C de configuration Courante de même dimension, s'obtiennent, à toute pompe (à Shadoks) si Y est la matrice des 9 transformations, par l'expression APL:

```

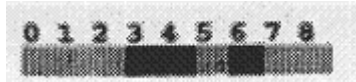
(( , C ≠ B) ≠ . ^ Y ≠ . ^ Y ≠ . ^ Y) / 1 9
3 4 5

```

à comme pour l'exemple :



Si DD est la différence linéarisée: , $C \neq D$ le masque-mystère



est donné par: $DD \text{ } PB \text{ } Y \text{ } PB \text{ } Y \text{ } PB \text{ } Y$.

Mais on pourrait aussi transformer le puzzle en puzzle fractal, car $G9$ est un géniton (voir [3], p. 37 ainsi que les références en [4]).

G_9 a la propriété d'être la parité du triangle de Pascal P_9 ;

$+P_9 \leftarrow \phi S \circ \cdot ! S \leftarrow \iota_9$

1	1	1	1	1	1	1	1	1
8	7	6	5	4	3	2	1	0
28	21	15	10	6	3	1	0	0
56	35	20	10	4	1	0	0	0
70	35	15	5	1	0	0	0	0
56	21	6	1	0	0	0	0	0
28	7	1	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0

Le cube matriciel de G_9 est une matrice-unité, comme pour tous les génitons d'ailleurs, sous-matrices du triangle fractal de Sierpinski INFINI, matrice universelle de parité du triangle de Pascal lui-même développé à l'infini...

Chaque ligne de G_9 peut se redimensionner aussi en matrice de dimension 3 3. On aurait un nouveau puzzle avec 9 transformations autres que celles proposées en [1] OU [2]. Alors, remplaçons Y par G_9 ; les 9 transformations sont aussi indépendantes, mais la matrice inverse de G_9 est alors son carré (et non plus son cube), et aussi son symétrique par rapport à la seconde diagonale, en APL: $\phi \ominus G_9$ ou $\ominus \phi G_9$ au choix.

Dans ce nouveau puzzle, la formule magique sera:

$$DD \ PB \ G_9 \ PB \ G_9$$

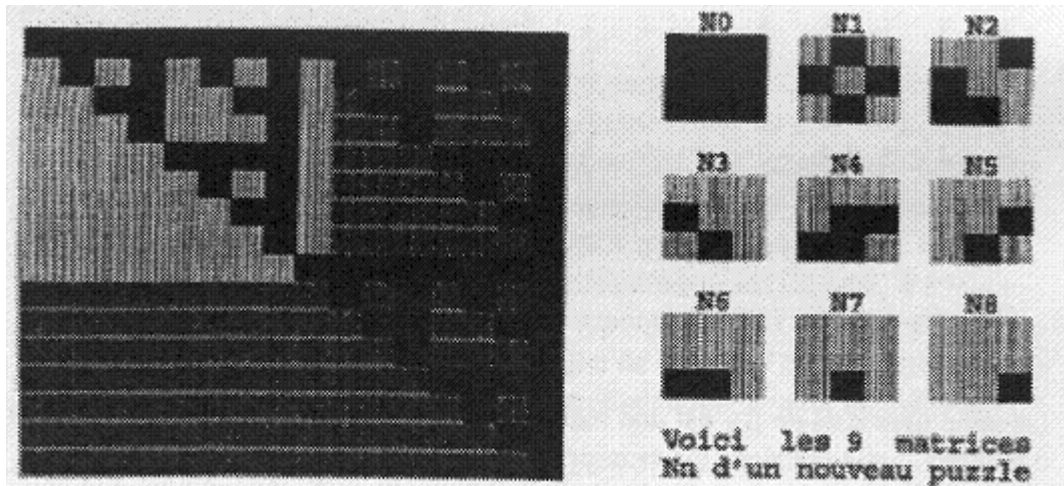
pour le masque-mystère des nouvelles matrices à appliquer.

Mais on peut encore faire mieux:

Si l'on asymétrise le puzzle, en considérant cette fois non plus les lignes de G_9 mais celles au choix de ϕG_9 ou de $\ominus G_9$, comme linéarisation des 9 matrices 3 3 de transformation, alors une NOUVELLE simplification apparaît: ϕG_9 et $\ominus G_9$ ont une matrice-unité pour carré: elles sont donc chacune leur PROPRE inverse; la formule se comprime encore algorithmiquement: $DD \ PB \ \phi G_9$

Que va-t-il rester si on continue à simplifier ainsi?

Il suffit de savoir que ϕG_9 représente une sous-matrice extraite des 9 premières lignes et des 9 premières colonnes de ϕG_{16} que voici sous forme semi-graphique avec contraste soutenu, en haut et à gauche, pour ϕG_9 :



Les transformations 0 2 3 4 5 6 8 permettent alors de passer de C à B avec ces nouvelles règles.

Chaque ligne L de $G16$ s'obtient aussi comme le résultat de l'itération de $L \leftarrow \neq \setminus L$ avec $L \leftarrow 16 \uparrow 1$ initialement, et les lignes de $G9$ ou de $\phi G9$ n'en sont qu'un morceau.

Mais, en se rapportant à d'autres publications comme [4], on s'aperçoit que le produit $S \cdot PB \cdot \phi G$ d'une séquence binaire S par une matrice de type ϕG est la transformée "hélicoïdale" de la séquence, alors que le produit $S \cdot PB \cdot \ominus G$ serait la transformée "cognitive" de S . Alors, on peut remplacer le produit matriciel PB (qui contenait encore un ET comme autre opération que \neq), par un algorithme ultra-rapide [5] ne faisant appel qu'à \neq TOUT SEUL, et dont le temps d'exécution n'est plus proportionnel à $N \times N$ si N est le rang de la matrice (ici 9) mais à N directement. On peut alors imaginer des puzzles de très grande taille, auprès desquels le carré 3x3 fait figure de goutte d'eau dans un océan de puzzles potentiels, TOUS aussi simples à résoudre, une fois qu'on a trouvé le truc grâce à APL et à la faculté que permet ce langage de raisonner en binaire directement, et de vérifier les idées avant de les propager.

A titre d'exemple, si HEL est une fonction fabriquant cette transformée hélicoïdale, le masque-mystère est:

```

9 ↑HEL 16 ↑DD
1 0 1 1 1 1 1 0 1

```

On peut retrouver ce même résultat avec $\neq \setminus$ itéré 16 fois sur DD considéré à l'envers, en ne prenant que 9 items sur les 16 accumulés dans T :

```

T ← 0 ρ S ← φ DD ◊ 'T ← T, ~1 ↑ S ← ≠ \ S' dO 16 ◊ 9 ↑ φ T
1 0 1 1 1 1 1 0 1      ρ dO est une fonction-boucle sans I

```

Il n'est alors pas plus difficile de considérer un puzzle de 4x4 soit 16 cases, avec 16 transformations issues des lignes de $G16$ alors repliées en matrices 4x4 également. Avec, cette fois 16 parités dans $DD \leftarrow$, $S \neq B$ le vecteur T contiendrait le masque-mystère, aussi à 16 bits, produit par l'expression:

```

'S ← ≠ \ S ◊ T [ I ] ← S [ I ]' dO I ρ T ← S ← DD

```

On peut signaler aux passionnés de puzzles que ces dernières expressions sont absolument équivalentes à ce que fait la transformation de Fourier sur un signal, mais, qu'en algèbre des parités, une fois les algorithmes comprimés, les résultats sont toujours justes et, évidemment, obtenus à la

vitesse de la lumière, avec \neq comme seule primitive de "calcul" encore nécessaire, toutes les autres pouvant aller au musée et y demeurer.

TOUTE MATRICE INVERSIBLE de toute taille (quel que soit le puzzle) se ramène toujours à un géniton ou à son symétrique ϕG qui est sa propre matrice inverse (ET aussi une forme de Jordan auto-diagonalisée), par combinaison linéaire; donc tous les puzzles sont des permutations de sous-matrices d'un seul et même puzzle, celui des parités de l'Univers dont $\neq \setminus$ serait (voir [4]) la loi d'évolution universelle ...

C'est ce que confirment de plus en plus de raisonnements et de preuves mathématiques, confrontées à diverses formulations récentes des "lois" décrivant les phénomènes physiques et biologiques.

Alors, il n'y avait pas de raison pour que l'inversion diabolique fasse exception à la règle déjà soupçonnée en 1989 comme le noyau unique de tous les algorithmes, après 10 ans de "compression algorithmique" effectuée, bien sûr, en APL.

Références (restreintes aux plus récentes)

- [1] J.J. Girardot, "L'inversion Diabolique", les Nouvelles d'APL (AFAPL), No 6, (mars 1993), p. 54-55.
- [2] D. Ziemann, "Exploring Magic", VECTOR (British APL Assoc.), Vol. 9, No 2 (Oct. 1992). Aussi: Bill Mc Lean, "MAGIC", Vector, Vol. 9, No 1 (Jul. 1992).
- [3] B. Mailhol, "Usage des Booléens", les Nouvelles d'APL, (AFAPL) No 6, (mars 1993), p. 25-44.
- [4] G.A. Langlet, "Towards The Ultimate APL-TOE (Theory of Everything)", APL92, St Pétersbourg, Russie; APL Quote Quad, Association for Computer Machinery, USA, (July 1992) No 22, vol. 1, p. 118-135. (& Réf. citées).
G.A. Langlet "Unitary Theory of Information", APL-CAM J. Bacus, "Part 1" : Vol. 13, No 2 (avr. 1991) p. 399- 432. "Part II" : Vol. 13, No 3 (juil. 1991) p. 709-743.
G.A. Langlet, "The Fractal Laws of Genetics", BIOMATH, (1992) Vol. XXX, No 118, p. 57-71.
G.A. Langlet "Le Principe de Moindre Action Généralisé" Colloque UITF, INSA, Rennes, Déc. 1992.
G.A. Langlet, "From the Vital Execute to Fractals and 5-fold Symmetry", VECTOR (British APL Assoc.), (Jan. 1993) Vol. 9, No 3, p. 91-96.
G.A. Langlet, "New Mathematics for the Computer", Tooi of Thought VIII, SIGAPL-New York, ACM (USA) (Jan. 1993) p. 1-31.
G.A. Langlet, "Symétries, Force et Phénomènes", APL-CAM Journal, Belgian Assoc. of Computer Users, [Ed.: J. de Kerf, Rooienberg 72, B-3570 Duffel] (Jan. 1993), Vol. 15, No 1, p.57-81.
G.A. Langlet, "From Risky Programming to the Baby- Computer", APL-CAM Journal, BACUS, (Jan. 1993), Vol. 15, No 1, p. 81-92.
- [5] G.A. Langlet, "Un Concours Paritonesque", APL-CAM J. BACUS, (Juil. 1992) Vol. 14, No 3, p. 407-411.

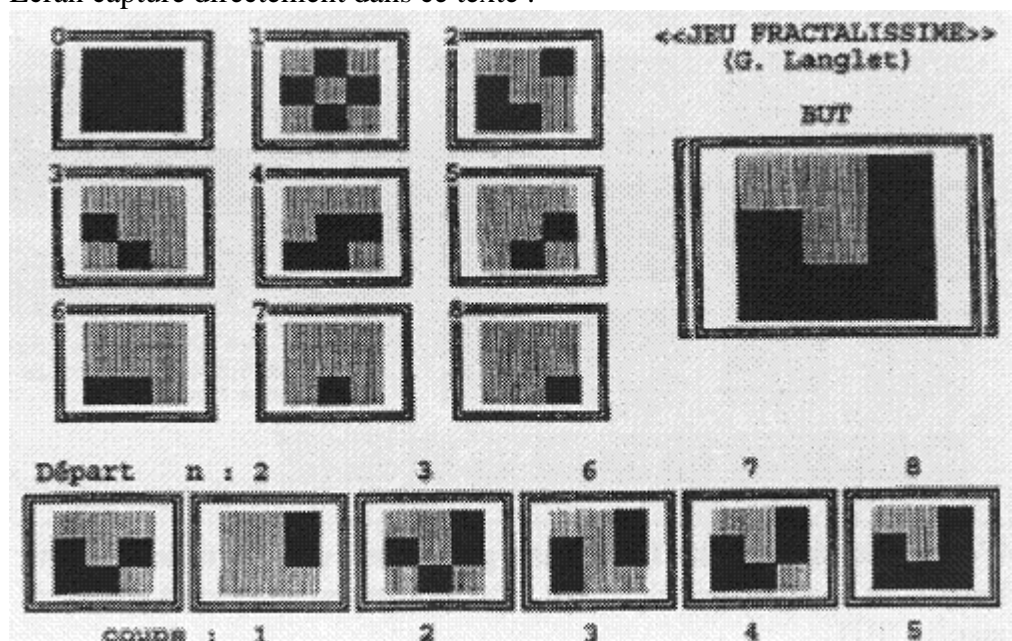
- [6] M. Dumontier, "Properties of Boolean Matrices Generated by Gray Functions", APL-CAM J. BAC US (Avr. 1992), Vol. 14, No 2, p. 302317.

Appendice.

Exemple testé sur PC-XT (APL*PLUS PC et II).

Le logiciel complet avec semi-graphique a moins d'un Koctet.

Ecran capturé directement dans ce texte :



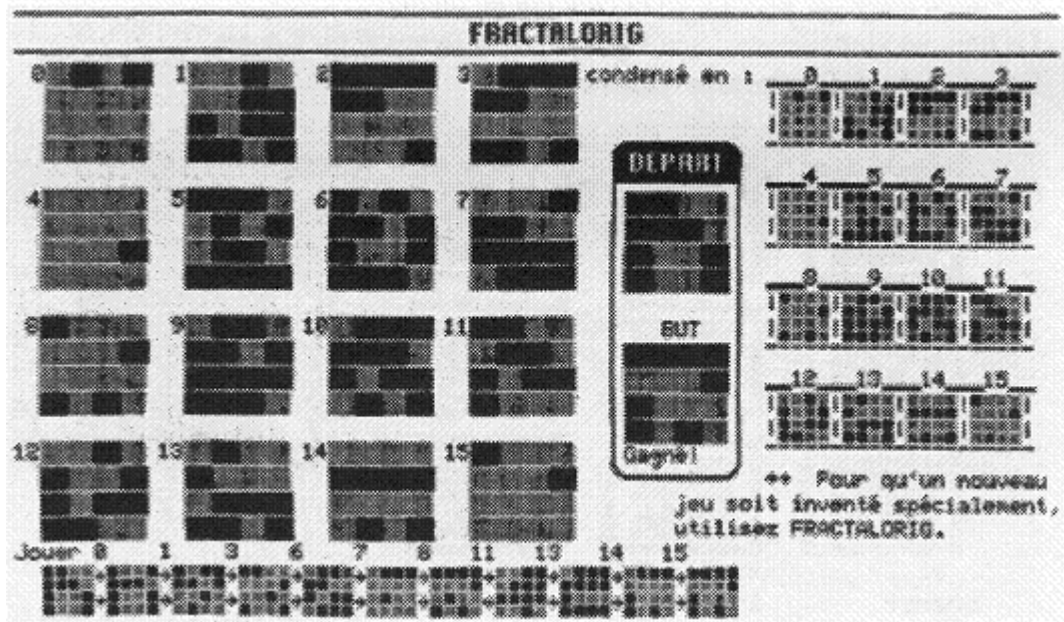
N.B. La fonction JEU est complète en UNE affreuse ligne d'APL... en programmation structurée sans branchement (\rightarrow), ni parenthèses, gérant le choix de la transformation à l'aide de la souris.

Les 9 matrices de transformation, linéarisées, forment la matrice appelée $\phi G9$ dans l'article. On peut changer aisément de règle en appliquant des combinaisons linéaires aux lignes de toute matrice déjà opérationnelle, toujours par \neq seul.

On peut envisager un programme fabriquant des puzzles à la chaîne en composant des ensembles de règles et en conservant ceux-ci quand la matrice est inversible (voir à ce propos l'étude de M. Dumontier [6]). Cette méthode s'appellera en jargon... le tirage au "XOR".

Post Scriptum

Exemple de nouveau jeu d'Inversion Diabolique 4x4 créé en une seconde sur Macintosh avec APL.68000, avec cette fois 16 matrices d'inversion de couleurs, numérotées de 0 à 15.



L'ordinateur fabrique les 16 masques graphiques, en deux tailles, tire au hasard une configuration de départ ainsi qu'une configuration à atteindre appelée BUT.

Puis il affiche ces deux configurations dans une fenêtre, en annonçant les numéros des masques qu'il conviendrait de jouer, pour gagner, ici 10 masques.

Il montre enfin les étapes intermédiaires sous forme graphique. Pour gagner contre la machine, il faudrait:

- a) jouer le premier
- b) ne JAMAIS commettre une seule erreur.