

Le compte est (souvent) bon

Michel Lafond^(*)

Résumé :

Dans cet article, on étudie divers aspects combinatoires du jeu « Le compte est bon ». Tous ceux qui ont pratiqué ce jeu se sont bien rendu compte qu'avec presque tous les tirages on obtenait le fameux « Le compte est bon ». Dans cet article, on montre que la probabilité d'obtenir le bon compte à partir d'un tirage aléatoire est d'environ 0,94. Les nombreux exemples cités permettent des travaux dans nos classes : calcul mental ou simplement divertissement mathématique.

Rappel historique tiré de Wikipédia.

Des Chiffres et des Lettres est un jeu télévisé créé le 4 janvier 1972 par Armand Jammot, reposant sur les compétences en calcul et sur la connaissance du vocabulaire des candidats. Diffusé sur la deuxième chaîne de l'ORTF, sur Antenne 2, France 2 puis sur France 3 depuis la rentrée scolaire 2006, il est aussi rediffusé par TV5. C'est la plus ancienne émission quotidienne toujours diffusée de la télévision française. Le jeu est actuellement présenté par Laurent Romejko, Arielle Boulin-Prat et Bertrand Renard, ces deux derniers contrôlant la validité des solutions proposées par les candidats.

Règle du jeu Le compte est bon

Les candidats disposent de 45 secondes pour effectuer sur leur écran tactile additions, soustractions, divisions ou multiplications afin de trouver exactement le nombre de trois chiffres proposé aléatoirement par le système informatique ou de s'en approcher le plus possible, à partir des six plaques de nombres proposées également aléatoirement par l'ordinateur.

Avant l'invasion de l'informatique, les six plaques servant aux calculs étaient tirées sans remise par un être humain, dans l'ensemble à 24 éléments :

PL = {1', 1'', 2', 2'', 3', 3'', 4', 4'', 5', 5'', 6', 6'', 7', 7'', 8', 8'', 9', 9'', 10', 10'', 25, 50, 75, 100}.

Autrement dit, les nombres de 1 à 10 étaient présents en double exemplaire dans « l'urne » alors que les quatre « gros nombres » ne figuraient qu'une fois.

Je distingue les deux plaques portant le 1 (par exemple) par 1' et 1'' afin de faciliter les dénombrements.

Après avoir tiré les six plaques, qu'il disposait sur un présentoir, l'opérateur affichait un nombre à trois chiffres à l'aide de trois rouleaux portant chacun les chiffres de 0 à 9 qu'il faisait tourner successivement.

(*) Dijon. mlafond001@yahoo.fr

Comme le règlement impose que le nombre N à obtenir soit compris entre 100 et 999, l'opérateur relançait le rouleau de gauche tant que celui-ci sortait le zéro.

Maintenant, c'est un logiciel qui tire tout cela, et nous supposons l'équiprobabilité :

– des combinaisons des 6 plaques de PL, et il y en a $\binom{24}{6} = 134\,596$.

– des nombres N à obtenir (entre 100 et 999). Il y en a 900.

Nous noterons $P = [p_1, p_2, p_3, p_4, p_5, p_6]$ la suite ordonnée des six valeurs des six plaques et N le nombre à obtenir.

Appelons **JEU** le couple (P, N) formé de la suite ordonnée des six valeurs, et du nombre N appelé **BUT**.

Il y a exactement $\binom{24}{6} \times 900 = 121\,136\,400$ jeux équiprobables.

Exemple : le JEU $([2, 2, 6, 10, 25, 100] ; 567)$ consiste à obtenir le BUT **567** à partir des nombres **2, 6, 10, 25, 100**, le 2 pouvant être utilisé deux fois. Attention, ici, la *valeur* 6 est indifféremment celle de la *plaque* 6' ou celle de la *plaque* 6''.

Définissons enfin deux sortes de jeux :

Les **jeux FAVORABLES** sont ceux pour lesquels il est possible d'obtenir le but en respectant les règles autorisées (décrites au paragraphe suivant) et les jeux **DÉFAVORABLES** (les autres).

Ainsi, le JEU $([2, 2, 6, 10, 25, 100] ; 567)$ est favorable puisque

$$6 \times 100 - 25 - 10 + 2 = 567.$$

Quelles sont les opérations autorisées ?

Le candidat a le droit d'effectuer jusqu'à cinq opérations (additions, soustractions, divisions ou multiplications) à partir des valeurs de P ou des résultats intermédiaires précédemment obtenus. Il n'est pas obligé d'utiliser toutes les plaques, mais chacune ne doit être utilisée qu'une seule fois.

Attention, les divisions éventuelles doivent être exactes (avec reste nul), autrement dit, il n'a pas droit aux fractions non entières.

Le but N étant strictement positif et l'obtention d'un zéro comme résultat intermédiaire étant inutile, on peut imposer que chaque résultat intermédiaire soit strictement positif, et je suppose que ce point est précisé dans le règlement.

Enfin le candidat ne peut opérer que sur deux entiers à la fois, ceci afin de faciliter le contrôle en temps réel par les organisateurs. Ainsi, toute solution doit être décomposable en une suite de calculs du type :

$$X * Y = Z.$$

Cela permet aussi aux nombreux téléspectateurs de suivre pas à pas le calcul.

Exemples : à partir de [2, 2, 6, 10, 25, 100] on peut obtenir :

$$129 \text{ par } 100 + 25 = 125 \quad 2 \times 2 = 4 \quad 125 + 4 = \mathbf{129},$$

$$108 \text{ par } 100 + 6 = 108 \quad 106 + 2 = \mathbf{108}.$$

Mais $100 + 6 + 2 = 108$ serait en principe refusé.

D'ailleurs l'addition en tant qu'opération *binaires* ne peut opérer en toute rigueur qu'avec deux opérands, même si tout le monde use de l'associativité.

Par contre 450 serait refusé si la solution est présentée ainsi :

$$25 + 2 = 27 \quad \frac{27}{6} = \frac{9}{2} \quad \frac{9}{2} \times 100 = \mathbf{450}.$$

Dans le cas présent, et c'est vrai la plupart du temps, l'usage des fractions ne s'impose pas, et ici on peut obtenir 450 simplement en respectant la règle avec

$$(10 - 6) \times 100 + 2 \times 25$$

à condition de décomposer le calcul en quatre étapes :

$$10 - 6 = 4 \quad 4 \times 100 = 400 \quad 2 \times 25 = 50 \quad 400 + 50 = \mathbf{450}.$$

On peut aussi écrire :

$$25 + 2 = 27 \quad 27 \times 100 = 2\,700 \quad 2\,700 / 6 = \mathbf{450}.$$

Avant de rentrer dans le vif du sujet, effectuons deux remarques :

1) Le jeu ([1, 2, 25, 50, 75, 100] ; 383) est défavorable (voir preuve plus loin), c'est-à-dire qu'on ne peut pas obtenir 383 à partir des plaques en respectant la règle, alors que mathématiquement il est exact d'écrire :

$$50 + 2 = 52 \quad 52 / 25 = \frac{52}{25} \quad \frac{52}{25} + 1 = \frac{77}{25} \quad \frac{77}{25} \times 100 = 308 \quad 308 + 75 = \mathbf{383} !$$

Mais ce genre de contre-exemple est très rare.

2) Le jeu $J_{100} = ([p_1, p_2, p_3, p_4, p_5, 100] ; 100)$ est un cas particulier si le candidat annonce fièrement au bout de ses 45 secondes « Le compte est bon » en proposant simplement comme solution « 100 ». Cela pose un problème, puisqu'il n'y a aucun calcul du type $X * Y = Z$ dans sa solution.

Mais c'est un faux problème car il se trouve que *tous les jeux* $J_{100} = ([p_1, p_2, p_3, p_4, p_5, 100] ; 100)$ sont favorables en respectant scrupuleusement la règle.

Ce n'est guère étonnant car :

Dès que deux valeurs x et $(x + 1)$ sont consécutives dans P on a la solution

$$(x + 1) - x = 1 \quad 1 \times 100 = 100$$

qui fonctionne et dès qu'une valeur y (entre 1 et 10) est présente deux fois dans P, on a la solution

$$y / y = 1 \quad 1 \times 100 = 100$$

qui fonctionne. Les autres cas comme :

$([p_1, p_2, p_3, 25, 75, 100] ; 100)$ avec la solution

$$25 + 75 = 100$$

ou $([2, 4, 6, 8, 10, 100] ; 100)$ avec la solution

$$2 + 4 = 6 \quad 6 / 6 = 1 \quad 1 \times 100 = 100$$

ou pour tout x , $([2, 5, 7, 25, x, 100] ; 100)$ avec la solution

$$7 - 5 = 2 \quad 2 + 2 = 4 \quad 4 \times 25 = 100$$

et les autres permettent systématiquement l'obtention de 100 par une ou plusieurs opérations *binaires*.

Comment rédiger un algorithme pour décider si un jeu est favorable ou non ?

Soit $J = ([p_1, p_2, p_3, p_4, p_5, p_6] ; N)$ un jeu quelconque.

En cours d'essais, on a éventuellement des résultats intermédiaires qui, avec les plaques restantes fournissent une liste d'au moins deux données (sinon l'essai est terminé) avec lesquelles il faut décider si N est accessible ou non.

Le travail (T) à faire est le suivant :

On dispose d'une liste d'au moins deux entiers non nuls $L = [x_1, x_2, x_3, \dots, x_n]$ et d'un entier N (le but qui ne varie pas).

Il faut impérativement pour chaque couple (x_i, x_j) avec $1 \leq i < j \leq n$:

a) Effectuer les quatre calculs :

$$z_1 = x_i + x_j \quad z_2 = |x_i - x_j| \quad z_3 = x_i \times x_j \quad z_4 = \max(x_i, x_j) / \min(x_i, x_j).$$

Remarques : la valeur absolue dans z_2 évite le passage inutile par les négatifs, mais ce calcul ne doit être effectué que si $x_i \neq x_j$ pour éviter le zéro ; $\min(x_i, x_j)$ n'est pas nul puisque aucun calcul intermédiaire n'est nul ; par contre z_4 doit être entier pour la suite

b) Pour chacun de ces résultats z , il faut tester si $z = N$.

Si c'est le cas, la recherche est terminée avec comme réponse OUI (Le jeu est favorable).

Sinon, il faut composer la nouvelle liste L' avec les termes de L privés de x_i et x_j qui ont été utilisés, mais augmentée de z .

c) On est alors confronté à un travail presque identique à (T), la seule différence étant la liste qui est L' au lieu de L (avec un terme en moins). Le travail aura donc une fin puisqu'on ne fait rien avec une liste d'un seul élément.

À la fin du travail, s'il n'y a pas eu arrêt avec la réponse OUI, c'est que le jeu est défavorable.

Il peut sembler gênant que le travail (T) renvoie à un travail de même nature, mais en fait (T) dépend d'un argument L (la liste sur laquelle il opère) et, si on regarde bien, (T) (L) se décompose en (au plus) quatre sous-travaux avec des listes plus petites. Les langages informatiques actuels permettent aisément ce genre de procédures nommées récursives.

Combien de temps met mon ordinateur pour effectuer un travail (T) ?

Ce temps varie de 0 à 20 secondes. Le pire des cas est lorsque la réponse est NON (jeu défavorable) car alors tous les calculs possibles ont été effectués.

Exemple. Avec le tirage de plaques $P = [1, 2, 25, 50, 75, 100]$, (T) met :

- environ 0,5 seconde pour trouver que 351 ou 352 sont réalisables,
- environ 8 secondes pour trouver que 365 est réalisable,
- environ 20 secondes pour trouver que 382 ou 383 ne sont pas réalisables.

Le temps moyen d'un travail est d'environ 5 secondes avec mon programme. Un spécialiste ferait évidemment mieux en optimisant l'algorithme.

Quelle est la probabilité de gagner ?

Autrement dit, quel est le pourcentage de jeux favorables parmi les 121 136 400 jeux équiprobables ?

Si l'on examinait chacun des 121 136 400 jeux équiprobables afin de compter le nombre de jeux favorables, cela prendrait de l'ordre de 600 000 000 secondes soit pas loin de 20 ans !

On peut procéder plus rapidement ainsi :

Pour chacune des $\binom{24}{6} = 134\,596$ listes de six plaques : $P = [p_1, p_2, p_3, p_4, p_5, p_6]$,

on utilise un tableau binaire POSS(100), POSS(101), ..., POSS(999) défini par POSS(N) = 1 si le jeu (P, N) est favorable, et POSS(N) = 0 sinon. Ce tableau est initialisé identiquement à 0.

On examine ensuite toutes les opérations possibles sur les six plaques de P. C'est-à-dire que le travail (T) défini au paragraphe précédent est effectué avec la liste P comme argument, sauf qu'ici le but N est indéterminé. Par contre, à chaque résultat intermédiaire obtenu, disons $X * Y = Z$, on pose POSS(Z) = 1 à condition évidemment que Z soit compris entre 100 et 999.

À la fin, c'est à dire au bout d'environ 20 secondes, le tableau POSS donne tout ce qu'il est légalement possible d'obtenir à partir des plaques P.

Ce qui nous intéresse est le *nombre de jeux favorables* associés aux plaques P, et

ce nombre est, par définition du tableau POSS, $S = \sum_{i=100}^{i=999} \text{POSS}(i)$.

Il suffit alors de faire varier P de toutes les façons possibles (soit 134 596), de sommer le nombre des jeux favorables S associés, et de diviser le tout par le nombre total de jeux possibles équiprobables (121 136 400) pour avoir la probabilité désirée.

L'utilisation du tableau POSS fait gagner un facteur temps d'environ 900 puisque 900 buts sont testés avec le même travail qui ne testait qu'un seul but auparavant. La durée des calculs serait de l'ordre d'une semaine.

Le calcul effectif de la probabilité.

Pour diminuer encore le temps de calcul, je procède ainsi :

Soit par exemple le tirage de plaques $P = [2, 7, 8, 25, 50, 75]$.

La valeur 2 est celle de la plaque 2' ou celle de la plaque 2''.

La valeur 7 est celle de la plaque 7' ou celle de la plaque 7''.

La valeur 8 est celle de la plaque 8' ou celle de la plaque 8''.

Seules les valeurs sont importantes pour les calculs, alors que le tirage des valeurs $[2, 7, 8, 25, 50, 75]$ est obtenu $2^3 = 8$ fois. Évidemment, un seul tirage sur les 8 sera testé.

Plus généralement, appelons **POIDS** du tirage de plaques $P = [p_1, p_2, p_3, p_4, p_5, p_6]$ le nombre 2^k où k est le nombre de « petites valeurs », c'est-à-dire inférieures à 25 et présentes une seule fois dans P .

Pour faciliter l'exploration en fractionnant le calcul, j'étudie séparément les tirages de plaques selon le nombre :

- de valeurs « simples » (inférieures à 25 et non doublées) présentes dans P ,
- de valeurs « doubles » (inférieures à 25 mais doublées) présentes dans P ,
- de valeurs « fortes » (supérieures à 25) présentes dans P .

Il y a 14 cas dont les conclusions sont reportées dans le tableau ci-dessous.

Les valeurs q_i ont été trouvées par ordinateur, avec 14 programmes presque identiques, ne différant que dans la manière de générer les tirages de plaques.

Par exemple, la signification des termes de la ligne 8 de ce tableau est :

Les tirages de plaques contenant deux valeurs simples, une valeur double et deux valeurs fortes comme $[3, 6, 6, 8, 50, 100]$ sont au nombre de 2160 :

$$\binom{10}{2} = 45 \text{ paires de valeurs simples parmi } 10,$$

que multiplie

$$\binom{8}{1} = 8 \text{ valeurs doubles parmi les } 10 - 2 = 8 \text{ valeurs restantes inférieures à } 25,$$

que multiplie

$$\binom{4}{2} = 6 \text{ paires de valeurs fortes parmi les } 4.$$

Le poids de ces tirages est $2^2 = 4$ car il y a deux plaques « 3 » et deux plaques « 8 ».

Sur tous les tirages de ce type, il y a au total 1 860 573 jeux favorables (lorsque le but varie), qui, compte tenu du poids 4, contribueront pour $1\,860\,573 \times 4 = 7\,442\,292$ au nombre total de jeux favorables.

Le programme que j'ai utilisé (avec Maple) est donné en annexe.

nombre de valeurs simples	nombre de valeurs doubles	nombre de valeurs fortes	Nombre de tirages (en valeurs) t_i	Poids p_i	Nombre de jeux favorables (en valeurs) q_i	Nombre de jeux favorables $p_i q_i$
6	0	0	$\binom{10}{6} = 210$	64	173 939	11 132 096
5	0	1	$\binom{10}{5} \times \binom{4}{1} = 1\,008$	32	903 164	28 901 248
4	0	2	$\binom{10}{4} \times \binom{4}{2} = 1\,260$	16	1 127 558	18 040 928
4	1	0	$\binom{10}{4} \times \binom{6}{1} = 1\,260$	16	928 915	14 862 640
3	0	3	$\binom{10}{3} \times \binom{4}{3} = 480$	8	416 657	3 333 256
3	1	1	$\binom{10}{3} \times \binom{7}{1} \times \binom{4}{1} = 3360$	8	2 920 023	23 360 184
2	0	4	$\binom{10}{2} \times \binom{4}{4} = 45$	4	36 976	147 904
2	1	2	$\binom{10}{2} \times \binom{8}{1} \times \binom{4}{2} = 2160$	4	1 860 573	77 442 292
2	2	0	$\binom{10}{2} \times \binom{8}{2} = 1\,260$	4	801 173	3 204 692
1	1	3	$\binom{10}{1} \times \binom{9}{1} \times \binom{4}{3} = 360$	2	277 314	554 628
1	2	1	$\binom{10}{1} \times \binom{9}{2} \times \binom{4}{1} = 1\,440$	2	1 148 697	2 297 394
0	1	4	$\binom{10}{1} \times \binom{4}{4} = 10$	1	6 789	6 789
0	2	2	$\binom{10}{2} \times \binom{4}{2} = 270$	1	207 662	207 662
0	3	0	$\binom{10}{3} = 120$	1	62 546	62 546
						113 554 259

On vérifie que le nombre total de tirages équiprobables de plaques est bien

$$\sum_{i=1}^{i=14} p_i t_i = 134\,596 = \binom{24}{6}.$$

La probabilité du succès est donc $\frac{\sum_{i=1}^{i=14} p_i q_i}{\binom{24}{6} \times 900} = \frac{113\,554\,259}{12\,113\,640\,000} \approx 0,937$ comme

annoncé au début.

Quelques cas remarquables.

La probabilité conditionnelle varie beaucoup selon la ligne du tableau précédent :

– Ainsi, pour les tirages de plaques contenant six valeurs simples (sans double ni

valeur forte) la probabilité du succès n'est que de $\frac{62\,546}{120 \times 900} \approx 0,579$ [dernière ligne

du tableau].

– Le pire des cas étant le tirage de plaques [1, 1, 2, 2, 3, 3] avec lequel on ne peut obtenir aucun entier entre 100 et 999 ! Ce cas est unique.

– À l'opposé, pour les tirages de plaques contenant cinq valeurs simples et une valeur

forte, la probabilité du succès est de $\frac{903\,164}{1\,008 \times 900} \approx 0,996$ [deuxième ligne du tableau].

Parmi ces « bons tirages » il y a des tirages de plaques qu'on pourrait qualifier d'UNIVERSELS en ce sens qu'il permettent de réaliser tous les entiers de 100 à 999 (et souvent beaucoup plus).

En voici quelques-uns :

$$[2, 3, 4, 5, 6, 75] ; [5, 6, 7, 8, 9, 25] ; [7, 8, 9, 10, 25, 75] ; \\ [5, 8, 9, 9, 75, 100] ; [6, 7, 10, 25, 50, 100], \text{ etc.}$$

Il y a plus d'un millier de tirages universels (sans compter les poids).

La probabilité pour qu'un tirage aléatoire de plaques soit universel est d'environ 0,18.

Enfin, des milliers de tirages de plaques sont quasi universels, en ce sens qu'ils permettent de réaliser tous les entiers de 100 à 999 *sauf un* !

En voici quelques-uns :

[4, 5, 6, 7, 9, 10]	permet de réaliser tous les entiers de 100 à 999 sauf 998.
[2, 2, 3, 9, 75, 100]	permet de réaliser tous les entiers de 100 à 999 sauf 989.
[7, 8, 10, 25, 50, 100]	permet de réaliser tous les entiers de 100 à 999 sauf 909.

EXERCICE : Trouvez le seul entier inaccessible (entre 100 et 999) à partir de [2, 3, 7, 8, 9, 10].

Le programme avec MAPLE.

Pour un tirage de plaques donné, ce programme sort la liste des nombres inaccessibles.

ajout:=proc(w) *procédure qui code 1 la possibilité d'atteindre le but w*

```
global poss:
if w>=100 and w<=999 then poss[w]:=1: fi:
end:
```

compte:=proc(l) *procédure qui opère sur les plaques de la liste l de toutes les façons possibles*

```
local e,i,j,k,x,y,z,z1,z2,l,l0:
global poss:
e:=nops(l):
if e>1 then
for i from 1 to e-1 do x:=l[i]: for j from i+1 to e do y:=l[j]:
  ll:=NULL:
  for k from 1 to e do
  if k<>i and k<>j then ll:=ll,l[k]: fi:
  od:
  z:=x+y: ajout(z): l0:=ll,z: compte(l0):
  if x<>y then z:=abs(x-y): ajout(z): l0:=ll,z: compte(l0): fi:
  z:=x*y: ajout(z): l0:=ll,z: compte(l0):
  z2:=max(x,y): z1:=min(x,y):
  if z2 mod z1=0 then z:=z2/z1: ajout(z): l0:=ll,z: compte(l0): fi:
  od:od:
fi:
end:
```

```
l:=[4,5,7,8,9,50]: la liste de plaques l à examiner
poss:=array(1..1000):
for i to 1000 do poss[i]:=0: od: mise à zéro du tableau POSS
compte(l): appel de la procédure principale
listeimp:=NULL: listage de tous les entiers inaccessibles
for i from 100 to 999 do if poss[i]=0 then listeimp:=listeimp,i: fi: od:
print (`sont impossibles :`,listeimp):
```