Qu'est-ce qu'un générateur de nombres au hasard?

Thierry Lambre(*)

Dans ce texte, nous proposons une description de quelques générateurs de nombres au hasard. Avant l'invention des ordinateurs, on utilisait des tables de nombres au hasard, notamment celles de la « *Rand Corporation* » ou celles de Rohlf & Sokal(1). Ces tables sont aujourd'hui aussi obsolètes que les célèbres « *Nouvelles Tables de Logarithmes* » de MM. Bouvart et Ratinet.

Quelle que soit sa forme, papier ou informatique, la production de nombres au hasard n'est pas une mince affaire car, outre la mise à disposition même de ces nombres, il s'agit également d'avoir une garantie sur la qualité du hasard fourni par ces nombres. Une liste (papier jusque dans des temps pas si lointains, informatique aujourd'hui) de nombres au hasard doit donc être validée par une série de tests de conformité parfois sophistiqués. Cette validation est un travail de professionnels pour lequel l'utilisateur est assez peu concerné sauf dans la confiance qu'il doit accorder aux listes de nombres produites, tout comme cet utilisateur accordait sans réserve sa confiance aux tables de logarithmes… Il est cependant sain d'avoir quelques idées sur la fabrication du hasard et sur l'évaluation de sa qualité.

Nous décrivons ici les générateurs les plus simples, qui peuvent quand même déjà s'avérer redoutablement efficaces. Nous donnons également quelques informations sur la façon d'évaluer leur fiabilité, ce qui est du ressort de l'informatique ou des statistiques.

Les mathématiques sollicitées pour la production du hasard par ordinateur sont diverses : corps finis, groupes cycliques, arithmétique (nombres premiers, congruence, réciprocité quadratique) ainsi que quelques outils de probabilités et statistiques (loi binomiale, loi faible des grands nombres). À la demande de la commission du bulletin (sans doute pour ne pas décourager le lecteur), toutes ces notions ne sont pas présentes dans ce texte(2).

Pour compléter ce texte, nous recommandons la lecture des deux fascicules *Statistiques au Lycée*, édités par l'APMEP cités en bibliographie, notamment les articles [P] et [L] ainsi que la lecture du chapitre 4.4 de [FLT].

Ce texte est organisé comme suit.

(*) Laboratoire de Mathématiques, UMR 6620 du CNRS, Université B. Pascal, 63177 Aubière Cedex. Courriel : thierry.lambre@math.univ-bpclermont.fr

(1) Nous expliquons plus bas comment ces tables ont été construites.

(2) Les curieux iront voir la littérature citée en bibliographie ou encore le texte [TL], qui, outre quelques démonstrations absentes ici, comporte divers compléments sur la simulation de la loi binomiale au travers d'une marche aléatoire dont l'origine remonte à Pascal et sa *géométrie du hasard*. Il serait peut-être dommage de se priver de cette promenade mathématique là.

1. Propriétés exigées d'un génrateur de nombres au hasard.
2. Les congruences de Lehmer (1948).
3. Suites géométriques modulo un nombre premier.
4. Le drôle de zèbre des calculatrices Texas Instruments.
5. Suites géométriques modulo une puissance de 2.
6. Tester la fiabilité d'un générateur de nombres au hasard.

1. Propriétés exigées d'un générateur de nombres au hasard.

Un générateur de nombres au hasard est un moyen informatique de simuler la loi uniforme sur l'intervalle]0; 1[ou la loi uniforme sur l'espace discret des entiers $\{1 ; \ldots ; N\}$. Un tel outil a de nombreuses applications en modélisation de phénomènes mathématiques, en probabilités et statistiques bien sûr, en simulation d'échantillonnages, mais aussi en traitement du signal, en synthèse d'images ou encore … pour les jeux informatiques(3).

Le principe de base d'un générateur de nombres au hasard est de fournir une suite $x_1 ; \ldots ; x_T$ de nombres « au hasard » pour T très grand. On exige de cette suite les propriétés minimales, mais un peu contradictoires, de *rapidité* et de *fiabilité*.

– *Rapidité* car les nombres doivent être produits rapidement. En effet, il n'est pas exceptionnel de souhaiter disposer d'un très grand nombre de nombres au hasard. Par exemple, dans la simulation d'un jeu de cartes, la simulation du battage des cartes revient à choisir l'une des 32! (ou pire 52!) permutations au hasard…
– *Fiabilité*, ce qui signifie que la suite doit modéliser la loi uniforme, sans biais majeur. Cette contrainte est très difficile à valider. Intuitivement, on souhaite obtenir une répartition très irrégulière (pour assurer le hasard), mais très régulière en moyenne (pour obtenir la loi uniforme). La répartition doit également simuler l'indépendance des différents tirages. Il faut en général investir de gros efforts dans la vérification de la qualité du hasard produit par un générateur donné. Cette validation de la qualité du hasard nécessite de nombreux tests de nature statistique.

Les générateurs de nombres au hasard sont assez fréquemment de la forme suivante :

On se donne un entier N, *le générateur fournit une suite de nombres entiers de l'intervalle* $[1 ; N - 1]$ *sous la forme* $u_{j+1} = f(u_j)$ *pour* $1 \leq j \leq N$ *où la fonction* $f : \mathbb{N} \to \mathbb{N}$ *ainsi que le premier terme* u_0 *sont très soigneusement choisis*.

La condition initiale u_0, appelée parfois graine ou germe (*seed* en anglais), doit avoir de grandes conséquences locales sans aucune conséquence globale. On ne peut rien prédire du générateur en fonction de la condition initiale sauf que, globalement, il modélisera toujours la loi uniforme quelle que soit la condition initiale retenue. Il faudra notamment exclure les conditions initiales qui pertubent le procédé.

On voit que la meilleure façon qu'on ait trouvé de simuler le hasard est d'en prendre le contre-pied en imaginant des mécanismes de production de nombres au déterminisme extrême. Ces processus de production de nombres sont si contraignants

(3) Chacun sait combien les jeux de hasard ont inspiré les pionniers du calcul des probabilités.

que les équations régissant ces phénomènes « hyper-déterministes » fournissent une bonne simulation du caractère imprévisible recherché dans la production automatique du hasard. D. Knuth, spécialiste d'algorithmique et créateur du logiciel de traitement de textes TEX résume plaisamment ce paradoxe : « *The moral of the story is that random numbers should not be generated with a method chosen at random*. »

2. Les congruences de Lehmer (1948).

Il semble que la technique *Middle Square* de J. Von Neumann (1946) soit l'une des premières tentatives de fabrication du hasard. Le principe en est hélas trop simple. Soit à trouver au hasard des nombres de 2N chiffres. On choisit un nombre u_0 de 2N chiffres. On calcule u_0^2, qui comporte 4N chiffres et u_1 désigne le nombre constitué des 2N chiffres du milieu de u_0^2. On obtient de la sorte une suite d'entiers de l'intervalle $[1 ; 10^{2N} - 1]$. En divisant par 10^{2N}, on obtient une suite de décimaux de $]0 ; 1[$. Ce générateur ne possède à peu près que des défauts : il dégénère très vite en une suite périodique, voire constante.

Les choses sérieuses commencent en 1948 avec l'apparition des générateurs de Lehmer, qui s'avèrent d'une toute autre efficacité.

Définition. *On appelle suite de Lehmer un suite* (u_n) *d'entiers définie par les conditions suivantes :*

$$u_{n+1} \equiv au_n + b \bmod \mathrm{N}$$

avec u_n *entier,* u_n *dans l'intervalle* $[0 ; \mathrm{N} - 1]$ *et où* N, *a, b et* u_0 *sont des entiers*.

Sous réserve de choisir avec discernement les entiers N, *a, b* et u_0, les suites de Lehmer fournissent de bons générateurs de nombres au hasard. Les outils mathématiques mis en jeu dans ces générateurs sont de nature arithmétique et algébrique.

Voici des situations montrant qu'il convient d'être vigilant.

Exercice.

a) *On considère la suite de Lehmer définie par* $u_{n+1} \equiv 25u_n + 16 \bmod 256$. *Que se passe-t-il pour* $u_0 = 10$ *?* $u_0 = 11$ *?* $u_0 = 12$ *?*

b) *On considère la suite de Lehmer définie par* $u_{n+1} \equiv 31\,415\,821\,u_n + 1 \bmod 10^8$ *et* $u_0 = 1$.

Regarder le dernier chiffre des premiers termes. Expliquer.

Remarquons que puisqu'on raisonne modulo N et, compte tenu de conventions retenues sur le choix des représentants de classes d'entiers, il n'y a que N valeurs au plus pour les termes de la suite de Lehmer.

Une suite de Lehmer est donc composée de T nombres en boucle parmi les N premiers entiers. Choisir les nombres N, *a* et *b*, de façon à obtenir le plus grand nombre T possible est un problème algébrique souvent accessible. Une fois ces deux étapes maîtrisées (choix de N, connaissance de la longueur T de la suite de nombres),

on est certes assuré de produire T nombres distincts parmi les entiers de l'intervalle [1 ; N – 1], mais rien de précis n'est dit sur la répartition de ces nombres relativement à la loi uniforme. Que ces nombres apparaissent au hasard ou pas est une toute autre histoire. Seule l'expérience et la pratique peuvent trancher. L'évaluation de la qualité du mélange est du ressort des statistiques. Comme nous l'avons déjà signalé, le générateur sera considéré comme fiable s'il résiste à un certain nombre de tests statistiques permettant d'armer que les nombres produits forment une représentation satisfaisante de la loi uniforme.

Si, dans une suite de Lehmer, nous supposons que $b = 0$, nous obtenons une suite géométrique modulo N. Nous nous concentrerons sur ces suites géométriques et traiterons plusieurs cas intéressants ou adaptés aux calculs sur ordinateurs : N premier, N premier de la forme $N = 2^n - 1$, N de la forme 2^n ou même proche de 2^n.

3. Suites géométriques modulo un nombre premier.

L'étude des suites géométriques modulo un nombre premier p est assez simple. Ceci explique peut-être que dans les bibliothèques d'ordinateurs, on trouve un grand nombre de générateurs de la forme

$$\begin{aligned} u_{n+1} &\equiv a u_n \bmod p, \\ u_0 &= 1. \end{aligned}$$

Citons quelques exemples.

• MAPLE : $p = 10^{12} - 11$, $a = 427\,419\,669\,081$.

• SAS : $p = 2^{31} - 1$, $a = 397\,204\,094$.

• Texas Instrument : $p = 2^{31} - 85$, $a = 40\,014$ et $p = 2^{31} - 249$, $a = 40\,692$. Nous verrons que le générateur des calculatrices TI, détaillé plus bas, nécessite l'utilisation *simultanée* de ces deux suites géométriques.

• Le générateur minimal standard : $p = 2^{31} - 1$, $a = 7^5$. Ce générateur fournit en boucle les $2^{31} - 2$ nombres entiers de l'intervalle $[1 ; 2^{31} - 2]$. Comme son nom l'indique ce générateur (introduit vers 1969 par Lewis et baptisé ainsi par Pack & Miller) a été validé par l'usage et est considéré comme le minimum exigible en terme de fiabilité relativement à la modélisation de la loi uniforme. Nous verrons plus bas que cette r éputation n'est pas usurpée.

Pour comprendre les raisons de ces valeurs numériques, un peu d'algèbre est nécessaire. En premier lieu, justifions le choix d'un nombre premier.

Soit p un nombre premier et désignons par $(\mathbb{Z}/p)^*$ le groupe multiplicatif des éléments inversibles de l'anneau $\mathbb{Z}/p$. Puisque p est premier, $\mathbb{Z}/p$ est un corps, le groupe $(\mathbb{Z}/p)^*$ possède $p - 1$ éléments. Le résultat ci-dessous est plus précis. Il s'avère utile pour construire des générateurs de nombres au hasard.

Théorème(4)**.** *Pour tout nombre premier, $(\mathbb{Z}/p)^*$ est un groupe cyclique d'ordre $p - 1$.*

(4) La démonstration est classique et se trouve dans tous les bons livres d'algèbre, par exemple [D], ouvrage ayant particulièrement pris en compte les moyens informatiques dont disposent les algébristes d'aujourd'hui.

Choisir un nombre premier pour les congruences présente donc deux avantages :
– La période T de la boucle est la plus longue possible : $T = p - 1$.
– Le groupe $(\mathbb{Z}/p)^*$ est cyclique. Il admet donc un générateur α, et $(\mathbb{Z}/p)^*$ est l'ensemble des valeurs de la suite géométrique (α^j) modulo p, c'est-à-dire

$$(\mathbb{Z}/p)^* = \{\alpha, \alpha^2, \ldots, \alpha^{p-2}, \alpha^{p-1} = 1\}.$$

Nous avons donc le résultat suivant.

Corollaire. *Soit p un nombre premier et soit α un entier de l'intervalle* $[1 ; p-1]$ *tel que* $\alpha = a \bmod p$ *soit un générateur de* $(\mathbb{Z}/p)^*$. *La suite géométrique modulo p définie par* $u_0 = 1$ *et* $u_{n+1} \equiv au_n \bmod p$ *fournit en boucle les* $p-1$ *entiers de l'intervalle* $[1 ; p-1]$, *chaque entier n'apparaissant qu'une seule fois dans chaque boucle*.

Commentaire : les choses ne sont jamais aussi simples qu'on pourrait le souhaiter. En effet, en choisissant $N = p$, nombre premier, on est assuré que le groupe multiplicatif $(\mathbb{Z}/N)^* = (\mathbb{Z}/p)^*$ est lui-même cyclique. On peut donc obtenir $p-1$ nombres « au hasard » en boucle. De plus, si α est un générateur de ce groupe cyclique, il suffit de prendre les puissance successives de α pour avoir la liste complète des $p-1$ éléments de ce groupe. Hélas ! Pour fabriquer ces nombres, il reste à trouver explicitement[5] un générateur α du groupe cyclique $(\mathbb{Z}/p)^*$. Et, malheureusement, le problème suivant est sans solution algorithmique satisfaisante a ce jour…

Problème. *Soit p un nombre premier et a un entier,* $1 < a < p$. *Décider si* $\alpha = a \bmod p$ *est un générateur du groupe cyclique* $(\mathbb{Z}/p)^*$.

Dans l'état actuel de nos connaissances, chaque situation est particulière. En voici quelques illustrations :
– Pour p premier, $p < 1000$, c'est un exercice instructif de trouver le plus petit entier positif a_p dont la classe modulo p est d'ordre $p-1$ dans $(\mathbb{Z}/p)^*$ (les mordus n'auront pas de difficulté à construire un petit programme).
– Reprenons les quatre exemples cités ci-dessus (Maple, SAS, TI, Minimun standard). S'il n'est pas difficile de vérifier que les nombres $10^{12} - 11$, $2^{31} - 1$, $2^{31} - 85$ et $2^{31} - 49$ sont premiers, il est plus délicat de montrer que, dans chaque cas, la quantité $\alpha = a \bmod p$ est un générateur du groupe $(\mathbb{Z}/p)^*$. Les seules démonstrations que j'ai pu construire nécessitent le petit théorème de Fermat, les carrés des corps finis et la réciprocité quadratique. On trouvera ces démonstrations détaillées dans [TL].

4. Le drôle de zèbre des calculatrices Texas Instruments.

Le générateur de nombres au hasard décrit dans ce paragraphe a pour origine une observation attentive de la fonction *Rand*( ) de la calculatrice TI Voyage. Il s'agit d'un drôle de zèbre, inventé par P. L'Écuyer en 1988, croisement de deux suites géométriques de Lehmer très judicieusement choisies. Une étude détaillée de ce

(5) Le tout n'est pas de savoir qu'il existe un banquier honnête, mais de le trouver…

générateur se trouve dans [TL].

Exercice[6]**.**
Considérons les nombres suivants $p = 2^{31} - 85$, $q = 2^{31} - 249$, $a = 40\,014$, $b = 40\,692$. *Alors les quantités* $\alpha = a \bmod p$ *et* $\beta = b \bmod q$ *sont des générateurs respectifs des groupes multiplicatifs* $(\mathbb{Z}/p)^*$ *et* $(\mathbb{Z}/q)^*$.

Le drôle de zèbre de l'Écuyer implanté sur les calculatrices TI se résume à la proposition suivante :

Proposition.
Posons $p = 2^{31} - 85$, $q = 2^{31} - 249$, $a = 40\,014$, $b = 40\,692$. *Soient* (s_n) *et* (t_n) *les suites de Lehmer définies par les conditions initiales* $s_0 \in [1\,;\, p-1]$, $t_0 \in [1\,;\, q-1]$ *et les relations de récurrence*

$$s_{n+1} \equiv as_n \bmod p \quad \text{et} \quad t_{n+1} \equiv bt_n \bmod q.$$

Soit (w_n) *la suite d'entiers de l'intervalle* $[1\,;\, p-1]$ définie par

$$w_n \equiv s_n - t_n \bmod p - 1.$$

Posons $\text{aléa}_n = w_n/(p-1)$. *Alors la suite* (aléa_n) *est une suite de nombres au hasard de l'intervalle* $]0\,;\, 1[$, *de période* $T = (p-1)(q-1)/2 \simeq 2.205\,10^{18}$, *soit une période de l'ordre de 2 milliards de milliards…*

Remarque. Les nombres p, q, a et b de la proposition n'ont pas été choisis au hasard :
– la taille de ces nombres a été choisie en vue de l'implémentation effective du générateur. Les nombres p et q comportent chacun 10 chiffres, les nombres a et b comportent chacun 5 chiffres. Les calculs des termes successifs des suites de Lehmer (s_n) et (t_n) ne mettront en jeu que des nombres comportant environ 15 chiffres, ce qui exclut les risques de dépassement de capacité ;
– les calculatrices TI travaillant sur 32 bits, il est naturel de choisir des nombres proches de 2^{32} ;
– la valeur $(p-1)(q-1)/2$ du ppcm de $p-1$ et $q-1$ est la plus grande possible pour des nombres impairs p et q. Ceci assure la plus grande périodicité possible du couplage de suites de Lehmer (s_n) et (t_n) ;
– les nombres p et q d'une part, a et b d'autre part, sont du même ordre de grandeur. On peut soupçonner que cela assure un mélange homogène dans la production des nombres au hasard mais cela mériterait d'être démontré car il n'est absolument pas clair que cette suite modélise correctement la loi uniforme.

Le générateur de nombres (aléa_n) de la proposition ci-dessus est installé sur les calculatrices TI sous la dénomination *Rand*( ). Voici un exemple pour se convaincre que l'instruction *Rand*( ) de TI coïncide bien avec la suite (aléa_n).
En posant $s_0 = 12\,345$, $t_0 = 67\,890$. Les premiers termes de la suite (aléa_n) sont :
$\text{aléa}_1 = 0,943\,597\,402\,493\,18$
$\text{aléa}_2 = 0,908\,318\,860\,975\,76$.
Initialisons à présent le générateur de nombres au hasard *Rand*( ) de la calculatrice

(6) Pour une solution, voir [TL].

par *RandSeed*0. Cette instruction a pour effet d'affecter les valeurs citées ci-dessus aux variables *Seed*1 et *Seed*2 : *Seed*1 = 12 345 et *Seed*2 = 67 890. Après cette initialisation, si on lance deux fois de suite l'instruction *Rand*(), on obtient
Rand( ) = 0; 943 597 402 492 13,
Rand( ) = 0; 908 318 860 974 75.
On a bien retrouvé aux erreurs d'arrondis près(7), les deux valeurs aléa$_1$ et aléa$_2$. Le hasard est donc bien démasqué sous une forme hyperdéterministe.

5. Suites géométriques modulo une puissance de 2.

Les congruences modulo une puissance de 2 sont bien adaptées aux ordinateurs car ceux-ci calculent en base 2. Les suites de Lehmer

$$u_{n+1} \equiv au_n \bmod 2^N$$

fournissent des générateurs de nombres au hasard de bonne qualité. En voici des exemples.

• $u_{n+1} \equiv 69\,069\,u_n \bmod 2^{32}$ (générateur de Marasaglia).

• Les tables de nombres au hasard de F.-J. Rohlf & R.-R. Sokal de 1969 ont été construites à partir de $u_{n+1} \equiv 5^{13}\,u_n \bmod 2^{325}$, (cf. [P], p. 182).

• APPLE : 2^{35}, $a = 1\,220\,031\,125$. La seule modification par rapport au cas précédent est le choix du générateur a.

• MATLAB : 2^{31}, $a = 16\,807$.

• $u_{n+1} \equiv 65\,539\,u_n \bmod 2^{31}$. Ce générateur ne semble pas produire un hasard de qualit é. Il est piquant de remarquer que ce dernier exemple, implanté jusque dans les ann ées 80 sur les ordinateurs IBM 360 ne donne plus satisfaction aux experts d'aujourd'hui. En outre, il convient de souligner que ce dernier exemple a été construit à partir de modifications minimes des valeurs des paramètres du générateur minimal standard qui, lui, donne toujours satisfaction. On a remplacé les valeurs $a = 7^5 = 16\,807$ et $N = 2^{31} - 1$ du générateur minimal standard par $a = 65\,539$ et $N = 2^{31}$. Ces minimes perturbations ont pourtant suffit à rendre inopérant un générateur fiable. On retiendra la leçon :

Il est extrêmement hasardeux de proposer un générateur sans l'avoir longuement testé pour le valider.

Pour comprendre la périodicité de ces générateurs, tous modulo une puissance de 2, il est nécessaire de maîtriser les suites géométriques modulo une puissance de 2. Un phénomène nouveau apparaît : contrairement au cas d'un nombre premier p, le groupe multiplicatif $(\mathbb{Z}/2^N)^*$ des éléments inversibles de l'anneau $\mathbb{Z}/2^N$ n'est pas cyclique. Mais il admet un gros sous-groupe cyclique pour lequel de plus on dispose d'une formule pour en trouver un générateur. L'énoncé précis est le suivant.

Théorème. *Soit* $N \geq 3$ *un entier.*

(7) Rappelons que dans tout calcul assisté par ordinateur, faire une confiance aveugle aux résultats affichés est une attitude d'une grande naïveté. Les erreurs d'arrondis sont tout sauf dérisoires. Nous n'aborderons pas ici ce point qui mériterait pourtant d'être longuement développé.

a) Les sous-groupes cycliques maximaux de $(\mathbb{Z}/2^N)^*$ *sont d'ordre* 2^{N-2}.
b) Soit a un entier impair et soit $\alpha = a \bmod 2^N$. *Pour que le groupe cyclique engendré par* α *soit maximal dans* $(\mathbb{Z}/2^N)^*$, *il faut et il suffit que* $a \equiv 3 \bmod 8$.
c) Le groupe $(\mathbb{Z}/2^N)^*$ *possède deux sous-groupes cycliques maximaux, d'ordre* 2^{N-2} *respectivement engendrés par* α *et* β *avec* $\alpha = 3 \bmod 2^N$ *et* $\beta = 5 \bmod 2^N$.

Pour une démonstration, voir [D].

Cet énoncé permet donc sans effort de construire des boucles de 2^{N-2} nombres parmi les 2^N premiers entiers.

Exemple. On considère la suite de Lehmer $u_{n+1} \equiv au_n \bmod 2^N$, $u_0 = 1$, u_n entier, u_n dans $[1 ; 2^N]$.
a) On suppose $a = 5$. Quelle est la période de la suite de nombres au hasard obtenus ?
b) On suppose $a = 3$. Quelle est la période de la suite de nombres au hasard obtenus ?
c) On suppose $a = 3$ et $u_0 = 3$. Quelle est la période de la suite de nombres au hasard obtenus ? Comparer la liste de nombres obtenus avec les deux précédentes listes.

• Qu'a-t-on gagné, qu'a-t-on perdu avec 2^N au lieu de p premier?
Perdu :
– Pour 2^N, $T = 2^{N-2}$, donc on n'obtient qu'un quart des entiers de $[1 ; 2^N]$ dans une boucle alors que pour p, p premier, $T = p - 1$, donc on obtient tous les entiers de $[1 ; p - 1]$ dans la boucle.
Gagné :
– Pour 2^N, on a sans effort un générateur du groupe multiplicatif $(\mathbb{Z}/2^N)^*$ car il existe une formule pour ce générateur. Ce n'est pas le cas pour un nombre premier p.
– Pour 2^N, les calculs en base 2 sont plus commodes et plus rapides car un ordinateur calcule en binaire. À partir d'une liste de nombres au hasard de $[1 ; 2^N]$, il est facile d'obtenir une liste de nombres au hasard de $]0 ; 1[$, par division par 2^N. Le grand avantage ici est que la division par 2^N est très rapide et n'introduit pas de biais dans les problèmes d'arrondis puisque diviser par 2^N en base 2 revient a décaler la virgule de N rangs. C'est là un avantage décisif pour les congruences modulo une puissance de 2.

6. Des générateurs additifs.

En dehors des suites de Lehmer, d'autres types de générateurs de nombres au hasard existent, notamment des générateurs additifs. Voici deux exemples ne nécessitant qu'une addition (ou une soustraction). Une addition en base 2 étant très rapide, il n'est pas étonnant que ces générateurs soient rapides. De plus, ces générateurs sont modulo une puissance de 2. La réputation de fiabilité de ces générateurs est solide. Ils ont subi un grand nombre de tests statistiques de manière satisfaisante. Le générateur de Mitchell-Moore

$$u_{n+1} \equiv u_{n-24} + u_{n-55} \bmod 2^N,$$

introduit vers 1958, ne nécessite qu'une addition. On peut démontrer qu'il possède une période multiple de $2^{55} - 1$. Il n'est pas possible avec les moyens actuels d'atteindre cette période, extrêmement longue.

Pour implanter ce type de générateur, il faut disposer d'une liste de 55 nombres aléatoires. Cette courte liste peut être obtenue par exemple à l'aide du générateur standard minimun.

Ces générateurs sont en quelque sorte une généralisation des suites de Fibonacci. Mais la suite de Fibonacci $u_{n+1} = u_n + u_{n-1}$ est un très mauvais exemple de générateur de nombres au hasard (même en raisonnant modulo un entier N). De manière un peu inattendue, les nombres 24 et 55 retenus ici pour le décalage d'indices suffisent à transformer la suite de Fibonacci en un excellent générateur de nombres au hasard. Ces décalages d'indices n'ont bien sûr pas été choisis au hasard et on serait mal inspiré de les modifier. Par exemple prendre $u_{n+1} = u_{n-5} + u_{n-55}$ au lieu de $u_{n+1} = u_{n-24} + u_{n-55}$ conduit à des périodes beaucoup moins grandes (15 533 si on raisonne modulo 2^{16}). L'étude théorique de ce générateur est assez subtile et sort du cadre de ces notes.

D. Knuth a proposé une variante de ce générateur. Cette variante du générateur de Mitchell-Moore donne d'excellents résultats. Il s'agit de

$$u_{n+1} \equiv u_{n-24} + u_{n-55} \bmod 2^N.$$

7. Tester la fiabilité d'un générateur de nombres au hasard.

Nous avons expliqué comment un ordinateur peut fournir rapidement une liste de T nombres tous distincts parmi les entiers 1, 2, ..., N. Cette possibilité d'un ordinateur, si intéressante soit-elle, ne peut en aucun cas prétendre être une simulation du hasard[8]. Il convient d'évaluer le désordre dans lequel cette liste apparaît.

L'une des qualités les plus importantes que doit posséder un générateur de nombres au hasard est de simuler l'indépendance des tirages successifs. Pour vérifier cette indépendance, on n'a pas encore trouvé d'autres moyens que de vérifier *a posteriori* sur les résultats fournis si cette indépendance est satisfaite ou pas… Ceci explique la présence de tests de validation.

Citons très succinctement quelques types de tests que doit subir avec succès un nouveau générateur de nombres au hasard[9].

• *Tests de fréquence*. Une manière grossière de vérifier l'homogénéité de la distribution des T nombres entiers sur l'intervalle [1 ; N] fournis par le générateur est de découper [1 ; N] en p parties de même longueur et de vérifier que sur chaque intervalle de longueur N/p, on trouve environ T/p nombres. Pour préciser ce qu'on entend par environ, on fait appel au test du χ^2, qui permet de décider si la répartition est suffisamment homogène à un seuil de confiance donné. On dit qu'on a testé la 1-distribution du générateur. En faisant des paquets de 2, 3, ... nombres, des tests plus sophistiqués sur la k-distribution ($k \geq 1$) des fréquences existent; ils consistent en la

(8) Une suite de nombres peut être très longue sans modéliser le hasard, la caricature étant la suite ordonnée des entiers !

(9) L'aspect allusif de la description des différents type de tests est dû à la faible pratique de l'auteur. Pour un panorama des méthodes les plus usuelles, voir [A] ; pour une présentation beaucoup plus détaillée, voir [FLT], chapitre 4.4., p. 180 à 198.

succession de tests de Diehard, disponibles sur les sites spécialisés. Si le générateur échoue à l'un de ces tests, il est rejeté.

• *Tests d'ordre*. Le nombre de montées ($u_{n+1} > u_n$) ainsi que la longueur des sous-suites extraites monotones sont des indicateurs de mélange que les experts savent analyser. Un autre test consiste a réordonner la suite dans l'ordre croissant, puis à calculer les différences successives $s_i = u_{i+1} - u_i$ et déterminer combien de fois un entier s apparait comme l'un des s_i. Là encore, analyser ces indicateurs permet de se prononcer sur la qualité du mélange.

• *Tests visuels*. Un moyen un peu rustique mais commode de déceler un biais dans une répartition supposée homogène est de faire tracer dans un carré $[0\ ;\ 1]^2$ un grand nombre de points à coordonnées choisies au hasard par le générateur. Le lecteur est invite a réaliser ce programme avec $n = 10^4$ et $n = 10^5$ pour obtenir une image de 10^4 ou 10^5 points du carré unité, choisis au hasard par le générateur de son instrument (par exemple la fonction *Rand*( ) de la TI Voyage).

• Le *test spectral* est un test d'origine visuel. Il est spécifique aux générateurs de Lehmer. Compte tenu du fait que u_{n+1} est fonction affine (modulo N) de u_n, les points de coordonnées $\left(u_n / \mathrm{N}, u_{n+1} / \mathrm{N}\right)_{0 \leq n \leq \mathrm{N}}$ se trouvent sur des segments de droites parallèles du carré unité. Notons $\mathcal{F}$ la figure obtenue. Le mélange sera d'autant plus homogène que la distance d entre deux segments parallèles voisins de $\mathcal{F}$ sera plus petite. La quantité $1/d$ est caractérisée par les nombres a, b et N intervenants dans la relation de récurrence $u_{n+1} \equiv au_n + b \bmod \mathrm{N}$. Cette quantité $1/d$, appelée précision du générateur de Lehmer, fournit donc un indicateur de mélange. Par exemple pour $b = 64$, $\mathrm{N} = 8\,505$, il est facile de voir sur l'écran de sa calculatrice que la précision du générateur défini par $a = 106$ est beaucoup plus petite que celle du générateur défini par $a' = 946$. Il sut pour cela de regarder les figures $\mathcal{F}$ et $\mathcal{F}'$ associées à ces deux valeurs dans des fenêtres adaptées. Le générateur défini avec $a = 106$ est donc de meilleure qualité que celui obtenu avec $a = 946$.

Pour une description plus précise de ces tests, nous ne saurions trop recommander au lecteur la lecture des 18 pages de [FLT], *Comment tester un générateur pseudo-aléatoire ?*, chapitre dans lequel 51 générateurs sont impitoyablement passés au crible d'une série de cinq tests(10). Leur bilan montre combien il faut être vigilant et proscrire toute adhésion à un argument d'autorité d'experts auto-proclamés. La moitié des 51 générateurs retenus est jugée peu fiable, tous les autres générateurs (sauf 4) présentent des biais mineurs souvent difficilement décelables, mais qu'il convient de connaître pour utiliser chaque générateur à bon escient en fonction du travail que l'on fait. L'un de ces auteurs, G. Fleury, m'a précisé que lors de ses études de recuits simulés, il avait été contraint d'éviter certains générateurs de Lehmer, dont il avait fini par percevoir les biais(11)…

(10) tests d'équirépartition, sériel, du poker, des permutations, des monotonies. Les conclusions de ces auteurs sont jointes sur un CD accompagnant leur ouvrage.

(11) Le recuit simulé intervient notamment dans le traitement informatique des images. La raison technique est la suivante : le recuit simulé s'appuie fortement sur l'indépendance des tirages successifs, ce qui peut être mal réalisé par un générateur de Lehmer.

Les quatre générateurs lauréats sur ces cinq tests successifs sont :
• Le générateur minimal standard $u_{n+1} \equiv 75u_n \bmod 2^{31} - 1$, dont le nom n'est donc pas usurpé.
• $u_{n+1} \equiv 3\,141\,592\,653u_n + 1 \bmod 2^{31}$ (il est plaisant de voir qu'une utilisation astucieuse des dix premières décimales de π fournit un hasard de qualité…).
• $u_{n+1} \equiv u_{n-6} + u_{n-31} \bmod 2^{31}$ (il s'agit d'un générateur additif).
• $u_{n+1} \equiv x_n + y_n + z_n \bmod 32\,3621$, avec
$x_n \equiv 157x_{n-1} \bmod 32\,363$,
$y_n \equiv 146y_{n-1} \bmod 31\,727$,
$z_n \equiv 142z_{n-1} \bmod 31\,657$.

Bibliographie.

[APMEP], *Statistiques au lycée*, brochures nº 156 (2005) et nº 167 (2007).
[A], D. Austin, Random numbers : nothing left to chance, *Am. Math. Soc.*, Monthly Essays on Mathematical Topics.
http://www.ams.org/samplings/feature-column/fcarc-index
[D], M. Demazure, *Cours d'algèbre : primalité, divisibilité, codes*, Cassini, 1997.
[En], A. Engel, *Mathématiques élémentaires d'un point de vue algorithmique*, Cedic, 1979, chap. 1 et 6.
[FLT], G. Fleury, Ph. Lacomme & A. Tanguy, *Simulation à événements discrets*, Eyrolles, 2006.
[K], D. Knuth, *The art of computing programming : seminumerical algorithms*, Adisson-Wesley, 1981.
[L], A. Ladureau, Utiliser une calculatrice comme générateur de hasard pour résoudre des problèmes de mathématiques, *in* [APMEP], brochure nº 167, 2007, p. 81-100.
[TL], Th. Lambre, Quelques aspects mathématiques de la simulation du hasard. à paraître sur le site de l'APMEP.
[L'], P. L'Écuyer, Efficient and portable combined random numbers generators, *Communication of the ACM*, 31, p. 742-749 et 774.
[Lar], P. Larbier, Produire des nombres « au hasard », 1994,
http://www.alrj.org/docs/algo/random.php
[P], B. Parzysz, Quelques questions à propos des tables et des générateurs de hasard, *in* [APMEP], brochure nº 156, 2005, p. 181-194.