par Hervé Lehning

Des algorithmes pour créer le hasard

Les algorithmes sont par nature déterministes. Pour une valeur donnée en entrée, ils fournissent toujours la même valeur en sortie. Pourtant, certains peuvent créer du hasard, ou plutôt le simuler. Un exemple très répandu est celui des générateurs de nombres pseudo-aléatoires. De tels algorithmes peuvent être utilisés pour calculer des intégrales multiples.

Pour simuler un phénomène fondé sur le hasard, on utilise des suites de nombres aléatoires c'est-à-dire fournis par le hasard. Les langages de programmation et les logiciels de calcul comme les tableurs offrent cette possibilité. Excel utilise une fonction nommée ALEA (pour *aléatoire*) et la plupart des langages de programmation une fonction nommée RAND (pour *random*).

Examinons la fonction proposée par Excel. Vous tapez =ALEA() dans une case et il renvoie un nombre entre 0 et 1 comportant huit chiffres après la virgule, ce qui revient, une fois la partie entière gommée, à un nombre compris entre 0 et 10^8. On obtient ainsi consécutivement des nombres ayant tout l'air d'être le fruit du hasard.

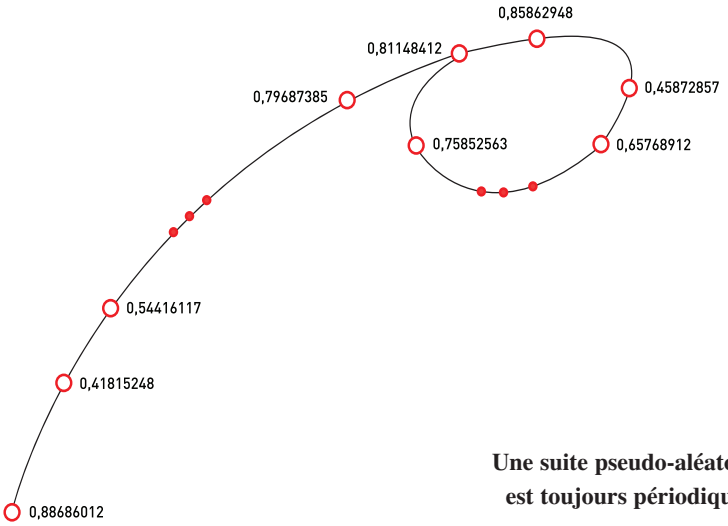
Le hasard est la mesure de notre ignorance.

Par exemple, voici une suite obtenue avec une ligne de dix instructions =ALEA() :

0,88686012	**0,41815248**
0,54416117	**0,59091058**
0,46098967	**0,73508295**
0,30939046	**0,79687385**
0,81148412	**0,85862948**

Suite de dix nombres obtenus avec la fonction ALEA.

D'où proviennent ces nombres ? Pas du hasard car rien n'est moins aléatoire que le fonctionnement d'un ordinateur. Même si l'utilisateur, et souvent même le concepteur, peut être surpris par certaines réactions de la machine, chaque opération suit un algorithme précis, son résultat est prédéterminé. Pas de hasard ici !



Une suite pseudo-aléatoire est toujours périodique.

Suite pseudo-aléatoire

Cependant, les suites produites respectent certaines règles des suites *véritablement* aléatoires. Tout d'abord, quand on augmente le nombre de termes de la suite, leur moyenne tend vers la moyenne attendue, 0,5 dans notre exemple. Bien entendu, il faut utiliser un grand nombre de termes pour la retrouver. La rencontrer trop tôt serait suspect…

Une suite *pseudo-aléatoire* est donc une suite de nombres entiers x_0, x_1, x_2, … prenant ses valeurs dans un ensemble M = {0, 1, 2, ..., $m-1$}. Le terme x_n ($n > 0$) est le résultat d'un calcul sur le terme précédent. Le premier terme x_0 est appelé le germe de la suite. Il la détermine donc entièrement ! L'ensemble M (les valeurs possibles de la suite) ayant m éléments, il est impossible que $m + 1$ termes de la suite soient distincts.

Ainsi, parmi les $m + 1$ premiers, deux sont forcément égaux. La règle de formation des termes de la suite implique qu'ils se répètent alors à l'identique ! Rien de moins hasardeux que cela ! Bien entendu, pour que ce phénomène ne soit pas gênant, on prend de très grandes valeurs de m. On a vu que le générateur d'Excel utilise $m = 10^8$. Les suites générées sont donc périodiques mais la période peut être très grande.
D'autre part, comme la suite dépend entièrement de son germe, si on ne le change pas, on retrouve toujours la même. Pour cette raison, les générateurs de nombres aléatoires introduisent à ce niveau un soupçon de vrai hasard. Par exemple, ils peuvent prendre comme germe les décimales de l'heure exacte où le programme a été démarré.

Les qualités d'un générateur idéal

Qu'attend-on d'un générateur de nombres pseudo-aléatoires ? La réponse à cette question conditionne la méthode à choisir car, dans ce domaine comme ailleurs, tout est affaire de compromis. Le jeu de dés utilise un générateur de nombres aléatoires, la simulation d'une centrale nucléaire

Simulation : la marche de l'ivrogne

Pour déterminer la probabilité d'un événement, une technique est de le simuler, ce qui nécessite d'utiliser un générateur de nombres pseudo-aléatoires.

Prenons l'exemple de la marche de l'ivrogne :

Un ivrogne sort de son bar préféré pour rentrer chez lui à trente mètres. Il n'a qu'un long trottoir rectiligne à suivre mais il est dans un tel état qu'il oscille aléatoirement à droite et à gauche à chaque pas sauf quand il est contre le mur. Il fait des pas d'un mètre de long et oscille de cinquante centimètres sur le côté à chaque pas. Le trottoir fait trois mètres de large.

Quelle est la probabilité qu'il tombe dans le caniveau ?

Le problème peut se résoudre par la théorie. Vous pouvez également simuler la marche de l'ivrogne. La difficulté est qu'il n'est pas facile de faire quoi que ce soit vraiment au hasard ! Si vous réalisez l'expérience physiquement, ne vous saoulez pas. D'abord, c'est mauvais pour la santé, ensuite il est plus simple d'utiliser une pièce de monnaie : pile, vous oscillez à droite, face, vous le faites à gauche. Vous la réalisez un nombre N de fois jusqu'à arriver chez vous ou tomber dans le caniveau. Vous comptez le nombre M de fois où vous finissez par terre. D'après la loi des grands nombres, si N est grand, la probabilité cherchée est proche de M/N. Il est fastidieux de réaliser toutes ces expériences physiquement même en utilisant un grand nombre d'ivrognes. Il est possible de réaliser une telle simulation en utilisant un tableur comme Excel. On rentre dans la première cellule : **=MAX(0;SI(ALEA()<0,5;-1;1))** puis dans la deuxième : **=MAX(0;SOMME(A1;SI(ALEA()<0,5;-1;1)))** ce que l'on recopie ensuite jusqu'à la cellule A30. Cela correspond à la marche d'un ivrogne. Il tombe dans le caniveau si le nombre 6 figure dans une des cellules A1 à A30 ce que l'on teste en écrivant dans la cellule A31 : **=SI(NB.SI(A1:A30;6)>0;1;0)**

Plus exactement, si notre ivrogne virtuel tombe dans le caniveau, la cellule A31 contient le nombre 1, sinon elle contient le nombre 0. Il nous reste à utiliser un grand nombre d'ivrognes virtuels en recopiant cette première colonne dans les suivantes. La moyenne des dernières lignes donne une approximation de la probabilité de tomber dans le caniveau. Pour affiner cette approximation, il est nécessaire de faire recalculer plusieurs fois la feuille utilisée. On trouve finalement que l'ivrogne à un peu moins d'une chance sur deux de finir dans le caniveau.Il est ainsi possible de simuler tout phénomène reposant sur le hasard.

aussi, mais les contraintes ne sont pas les mêmes. En général on privilégie trois critères.

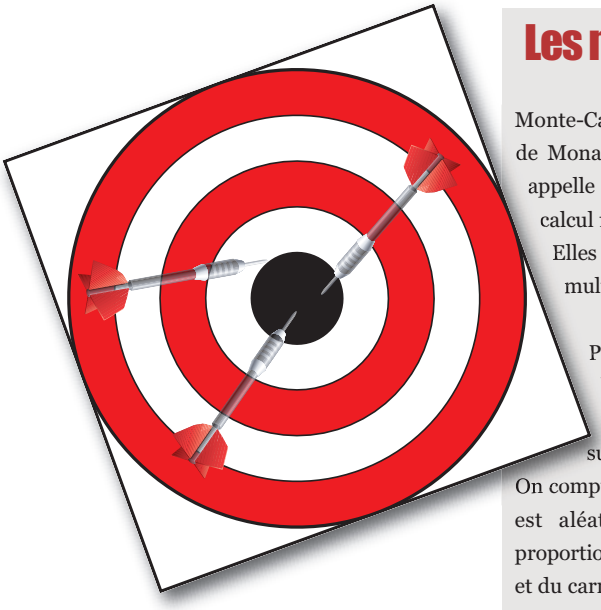
La vitesse : le calcul du nombre pseudo-aléatoire suivant doit être rapide. La simulation demande de générer parfois des millions de nombres.

La simplicité : une méthode très compliquée est très difficile à programmer et à tester.

La fiabilité de production des nombres : le programme ne doit pas « planter » quand il arrive sur certains nombres !

Les nombres produits ne doivent pas faire apparaître de suites logiques. Chaque nombre doit avoir à tout moment et quel que soit l'*historique* autant chance de sortir que les autres. Plus généralement, les théorèmes connus de probabilité doivent être vérifiés.

Dans la mesure où la suite proposée respecte toutes les lois connues des probabilités, on peut espérer qu'elle se comporte correctement dans les applications de simulation. Il ne s'agit pas d'obtenir vraiment du hasard. La question relève d'ailleurs plus de la philosophie que des mathématiques.



Les méthodes de Monte-Carlo

Monte-Carlo est connu pour l'utilisation que fait la principauté de Monaco du hasard pour s'enrichir. En son hommage, on appelle méthodes de Monte-Carlo toutes les méthodes de calcul fondées sur l'utilisation du hasard.

Elles sont couramment utilisées pour calculer des intégrales multiples.

Pour en donner une idée, voyons comment calculer π à l'aide... du hasard !

On tire de manière aléatoire des fléchettes (ou autres) sur une cible carrée dans laquelle on a inscrit un cercle. On compte ensuite le nombre d'impacts dans le disque. Si le tir est aléatoire, le nombre de coups dans le disque est proportionnel au rapport des surface du disque et du carré c'est-à-dire à $\pi / 4$.

Cette méthode est difficile à appliquer avec un tireur humain mais on peut l'utiliser en simulant le tir à l'aide d'un générateur de nombres aléatoires. C'est moins dangereux et plus sûr ! L'expérience montre que la convergence est assez lente.

Connaissant les théorèmes qui doivent être vérifiés, on en déduit une batterie de tests qu'une suite doit passer pour être considérée comme aléatoire ou plutôt pseudo-aléatoire puisque le hasard véritable est impossible à recréer. D'ailleurs, existe-t-il ?

Un exemple de générateur « historique »

Bien qu'on utilise de nos jours des suites plus sophistiquées, on a longtemps utilisé $m = 2\,147\,483\,647$ $(= 2^{32} - 1)$ et après avoir choisi un germe x en utilisant l'horloge de l'ordinateur, on calculait le reste de $16\,807\,x$ dans la division par $2\,147\,483\,647$ pour obtenir le suivant puis on recommençait.

Une telle méthode est très facile à programmer et fournit des suites du type :

321 635 411,	**510 013 178,**
1 184 247 469,	**768 771 087,**
1 474 038 857,	**799 717 807,**
1 904 519 323,	**1 012 503 126,**
479 619 854,	**146 4758 987...**

Cette suite passe tous les tests usuels et peut donc être considérée comme acceptable. Nous vous la recommandons si vous voulez fabriquer un générateur vous-même. Vous pouvez ainsi voir avec cet exemple que le hasard programmé n'a rien en commun avec le hasard. Si vous l'utilisez pour jouer aux dés avec des amis, vous pouvez toujours trouver un nombre à partir du précédent à condition d'être capable de calculer très vite et de tête. Mais ne dit-on pas que le hasard est la mesure de notre ignorance ? Après tout, le lancer de dés est aussi contrôlé par des équations connues. Si vous connaissez les conditions exactes du lancer, vous devez connaître le résultat.

H. L.