COURS D'INTRODUCTION À LA SIMULATION STOCHASTIQUE

FRANCK VERMET (EURIA)

1. Introduction

Ce cours est une introduction à la "simulation stochastique" ou "simulation Monte Carlo", que l'on peut définir comme étant l'ensemble des méthodes numériques de résolution de problèmes faisant usage de nombres aléatoires.

Pourquoi simuler sur ordinateur ? Pour résoudre numériquement des problèmes réels, que l'on ne sait pas résoudre analytiquement. Les méthodes d'analyse numérique, qui se développent depuis les années 1950, prennent aujourd'hui un essor considérable, l'augmentation très rapide des performances de calcul des ordinateurs permettant de mettre en oeuvre des algorithmes jugés beaucoup trop gourmands en temps de calcul il y a quelques décennies. Cependant, il faut se garder d'un excès d'optimisme : bien des problèmes, à priori simples, sont inaccessibles par les méthodes classiques, d'où aussi la forte effervescence de méthodes "multi-disciplinaires" (réseaux de neurones, logique floue, algorithmes génétiques, recuit simulé...). Mais cela dépasse le cadre de ce cours introductif.

Pourquoi utiliser des méthodes stochastiques ? L'aléa peut simplement être une astuce pout traiter un problème purement déterministe, ou bien provenir d'une modélisation stochastique d'un problème réel sur lequel on dispose d'une information lacunaire. Par exemple, on peut utiliser une méthode Monte-Carlo pour évaluer numériquement une intégrale multidimensionnelle $\int \dots \int_{[0,1]^d} f(x_1, \dots, x_d) dx_1 \dots dx_d$, que l'on peut exprimer comme la limite presque sure de la somme $\frac{1}{n} \sum_{i=1}^{n} f(U_i)$, si (U_i) est une suite de variables aléatoires indépendantes identiquement distribuées de loi uniforme sur $[0,1]^d$. Il s'agit dans ce cas d'un moyen de calcul adapté à un problème déterministe. Historiquement, la méthode Monte Carlo et son nom ont d'ailleurs été suggérés durant la seconde guerre mondiale au Laboratoire de physique de Los Alamos (U.S.A.) par le mathématicien S. Ulam pour évaluer des intégrales qui apparaissent dans la modélisation de réactions en chaîne nucléaires. Cette méthode, qui a été ensuite développée par Von Neumann, Metropolis et d'autres, existe donc depuis le début de l'informatique.
Une approche Monte-Carlo est utile aussi pour étudier le comportement d'une file d'attente, ou la dynamique d'une population, ou l'évolution génétique : dans ce cas, la stochasticité est intrinsèque, puisqu'il est impossible de connaitre à priori le comportement de chaque "individu". Et là encore, l'aléa peut être vu de deux manières. Prenons l'exemple de sociétés de fourmis : certains voient dans le comportement de l'individu fourmi, apparemment aléatoire, un caractère purement déterministe, mais caché. Mais depuis peu, le fait que l'aléatoire puisse être une composante non seulement intrinsèque mais également fonctionnellement pertinente de comportement individuel

Date: 13 novembre 2017.

constitue un pas essentiel vers la reconnaissance de l'importance du caractère fondamentalement non-déterministe de certains comportements. Ainsi, Ostet et Wilson (1978), par exemple, ont avancé l'idée selon laquelle ce caractère non-déterministe pourrait apporter une certaine robustesse à une société, en augmentant la probabilité qu'une tâche soit accomplie en toutes conditions. On peut aussi ici évoquer le "brassage génétique" de la reproduction sexuée.

Problèmes types usuellement traités par les Méthodes Monte Carlo :

- Simulation de modèles complexes issus de la physique mathématique et de la théorie des probabilités :
  - Modèles de physique statistique (ex. : le modèle ferromagnétique de Ising),
  - Modèles de réseaux de neurones (ex. : le modèle de Hopfield),
  - Modèles de polymères, de croissance de cristaux,
  - Rayonnement cosmique,
  - Percolation,
  - Automates cellulaires,
  - Modèles d'avalanche.

- Calcul mathématique :
  - Calcul d'intégrales multidimensionnelles,
  - Optimisation combinatoire (ex. : le problème du voyageur de commerce),
  - Discrétisation d'équations différentielles stochastiques.

- Simulation de fonctionnement et anticipation :
  - Mathématiques financières, actuariat (ex. : simulation de scénarios économiques, d'occurences de sinistres),
  - Files d'attente,
  - Centrales nucléaires (radioactivité),
  - Tests d'hypothèses extrêmes (ex. : cascades de pannes, fissures dans des structures architecturales),
  - Modèles sociologiques,
  - Propagation des épidémies,
  - Sociétés animales (ex. : les insectes sociaux).

2. Générateurs de nombres au hasard

Toute simulation Monte Carlo fait intervenir, par définition, des nombres au hasard. La question que l'on se pose est alors : comment générer une suite de nombres qui sont la réalisation d'une suite $X_1, X_2, \ldots, X_n$ de variables aléatoires indépendantes et de même loi ?
La méthode la plus "naturelle" est de trouver un phénomène physique bien modélisée par un processus de cette loi et d'identifier les valeurs successives mesurées de la grandeur physique avec la suite de nombres aléatoires cherchée. C'est cette méthode qui était utilisée avant 1940, quand on ne disposait pas d'ordinateur : les premières suites de valeurs aléatoires étaient alors générées par des dispositifs analogiques.

Exemples :
- Les fluctuations de tension aux bornes d'une résistance électrique chauffée par un courant stabilisé sont bien modélisées par un bruit blanc (loi normale).

- Les instants où des coups sont enregistrés par un compteur de rayonnement cosmique suivent une loi exponentielle.

A noter qu'en 1927, Tipett a conçu une suite de 40 000 nombres "tirés au hasard dans des tables de recensement", et dans les années 1940, en Angleterre, ERNIE, "the random number machine", produit des listes de nombres aléatoires à partir des tirages officiels des loteries. En 1955, la table RAND est une suite de 1 million de digits produits par un "bruit électronique". Mais ces suites physiquement construites ne sont pas satisfaisantes pour plusieurs raisons :
- non-reproductabilité lors d'un couplage direct entre un dispositif analogique et un ordinateur numérique, et lenteur de la procédure.
- Il faut stocker toute la suite de nombres pour pallier aux défauts précédents, or aujourd'hui, on a besoin de suites très longues.
- Problèmes de fiabilité et de la précision des mesures qui peuvent induire des biais et des corrélations.
- Le caractère uniforme des suites ainsi produites est invérifiable (ex. : les procédures utilisant les valeurs de l'horloge interne de l'ordinateur ont été abandonnées pour cette raison et la non-reproductabilité).
L'essor des ordinateurs dans les années 50 a conduit les scientifiques à préférer à ces méthodes analogiques des méthodes purement numériques (et à fortiori déterministes !) Un des premiers algorithmes est celui du "middle square" de Von Neumann : supposons que l'on veuille une suite de nombres décimaux à 4 chiffres. On part d'une valeur initiale quelconque, par exemple 8653. Alors $8653^2 = 74\mathbf{8744}09$. On prend les 4 chiffres du milieu et on recommence : $8653, 8744, 4575, 9306, \ldots$ On obtient une suite déterministe qui "parait" aléatoire.

Remarque 2.1. *en divisant chacun des nombres par 10 000, on obtient une suite dans* $[0, 1[$. *On souhaite souvent simuler une suite de variables aléatoires réelles de loi uniforme sur* $[0, 1]$ *(on verra bientôt pourquoi...), mais pour des raisons évidentes (représentation fini des réels sur ordinateur), on ne sait manipuler que des nombres ayant un nombre fini de décimales. Aussi, pour éviter les biais et corrélations dûs aux erreurs d'arrondi, on préfère simuler des v.a. de loi uniforme sur* $\{1, 2, \ldots, M\}$, *où* M *est grand (au plus de l'ordre de grandeur du plus grand entier codable par la machine).*

Tout en restant conscient des difficultés et interrogations qu'elle peut soulever (comment concilier aléatoire et déterminisme ?), nous définissons maintenant ce que nous entendrons désormais par une suite de nombres pseudo-aléatoires.

Définition 2.2. *On appelle* **générateur uniforme de nombres pseudo-aléatoires** *un algorithme fondé sur une valeur initiale* u_0 *et une transformation* D *qui produit une suite* $(u_i = D^i(u_0))$ *à valeurs dans* $[0, 1]$ *telle que* $(u_1, u_2, \ldots, u_n)$ *reproduit pour tout* n, *le comportement d'un échantillon uniforme i.i.d.* $(v_1, v_2, \ldots, v_n)$, *au sens où elle a les mêmes propriétés statistiques que l'échantillon uniforme.*

Pour valider une méthode de génération de nombres pseudo-aléatoires, il faut donc générer des échantillons et vérifier qu'ils satisfont à "tous" les tests statistiques connus. Bien entendu, ceci est purement théorique, car la liste des critères permettant de dire qu'une suite est aléatoire n'est pas exhaustive, et porte sur des échantillons de taille arbitrairement grande. Cela n'a pas de sens de dire qu'une suite donnée de 100 nombres est aléatoire : il peut exister un algorithme déterministe

la générant mais non détectable facilement. Nous pouvons distinguer deux grandes familles de tests statistiques, liées à deux propriétés fondamentales :
- la **loi** de l'échantillon (ex. le test de Kolgomorov-Smirnov),
- la propriété d'**indépendance** (absence de corrélations) (ex. le test du χ^2 d'indépendance).

2.1. **Générateurs par récurrences scalaires congruentielles.** Nous nous limiterons dans le cadre de ce cours aux générateurs par récurrences scalaires congruentielles : ils ont été pendant longtemps les plus utilisés de par leur simplicité et leur qualité. Prendre garde cependant que certains ne sont pas bons ! Ces générateurs sont du type :

$$x_0 \in \{1, \ldots, m\}, \quad x_i = (a\, x_{i-1} + c) \mod m, \quad \text{pour } i \geq 1.$$

Un générateur est donc défini par les trois valeurs entières a, c et m. Les nombres générés $(x_1, x_2, \ldots)$ sont des entiers dans $\{0, \ldots, m-1\}$. On obtient des nombres dans $[0, 1]$ en divisant les x_i par m. Cette suite est nécessairement périodique (puisque le nombre de valeurs possibles est au plus m), de période au plus m (en prenant $a = 1,\ c = 1$ par exemple). Il y a donc intérêt à choisir m grand (de l'ordre du plus grand entier codable par la machine), et a et c tels que la période soit la plus grande possible. Cependant, ce critère n'est pas suffisant : par exemple, le choix $a = 1,\ c = 1$ est-il judicieux ?

Le théorème suivant donne une solution complète au problème de maximisation de la période :

Théorème 2.3. *Le générateur "$x_i = (a\, x_{i-1} + c) \mod m$" a une période égale à m* **ssi** *les trois conditions suivantes sont vérifiées :*
1. c et m sont premiers entre eux,
2. $a - 1$ est multiple de p, pour tout facteur premier p de m,
3. si m est multiple de 4, alors $a - 1$ est multiple de 4.

On pourra trouver une démonstration de ce théorème dans le livre de Knuth, "The Art of Programming", Vol.2. Le premier exemple connu de ce type de générateurs utilisait les paramètres : $m = 2^{35},\ , a = 2^7$, et $c = 1$, et ne vérifiait donc pas les hypothèses de maximisation de la période. Ultérieurement, la valeur $a = 2^7 + 1$ a été proposée, qui les vérifie et donne de bonnes propriétés statistiques.

Un cas particulier, le plus utilisé de nos jours : $c = 0$. On sait que la période du générateur $x_i = a\, x_{i-1} \mod m$ est $< m$; par exemple on a plus précisement :

Théorème 2.4. *Le générateur "$x_i = a\, x_{i-1} \mod m$", avec $m = 2^\beta$ pour un entier $\beta \geq 4$, a une période (maximale) égale à $m/4$* **ssi** *les deux conditions suivantes sont vérifiées :*
1. x_0 et m sont premiers entre eux,
2. $a \mod 8 \in \{3, 5\}$.

Si $c = 0$, la valeur 0 ne doit pas apparaitre dans la suite des x_i, sinon elle dégénère $(0, 0, \ldots)$. Peut-on trouver des valeurs de a et m telles que les $m - 1$ autres valeurs apparaissent périodiquement ? La réponse est donnée par le théorème suivant.

Théorème 2.5. *si m est premier, la période du générateur "$x_i = ax_{i-1} \mod m$" est un diviseur de $m - 1$. Elle est égale à $m - 1$* **ssi** *m est premier et a est une racine primitive de m (i.e. $a \neq 0$ et $a^{\frac{m-1}{p}} \neq 1 \mod m$ pour tout diviseur premier p de $m - 1$).*

Remarque 2.6. *Par le petit théorème de Fermat, on a : m premier* $\Rightarrow a^{m-1} = 1 \mod m$*, pour tout* $a \neq km$*.*

Il peut s'avérer difficile de trouver les racines primitives. Mais si une telle racine est trouvée, toutes les autres sont obtenues par le théorème suivant.

Théorème 2.7. *si a est racine primitive de m premier, alors ceci est vrai pour* $a^k \mod m$*, pourvu que k et* $m-1$ *sont premiers entre eux.*

Il est à noter que ces générateurs sont en grande partie justifiés par des arguments issus de la théorie des nombres, la factorisation de $m-1$ en nombres premiers jouant un rôle important. Un exemple classique ayant de bonnes propriétés statistiques correspond au nombre de Mersenne $m = 2^{31} - 1$, qui est premier. On a : $m - 1 = 2.3^2.7.11.31.151.331$, et on vérifie que 7 est une racine primitive, donc $a = 7^5 = 16807$ également.
D'où le générateur :

$$x_0 \in \{1, \ldots, 2^{31} - 1\}, \;\; x_i = 16807\, x_{i-1} \mod (2^{31} - 1).$$

Code R : Voici une fonction R réalisant le générateur précédent.

```
generateur = function(x,n)
{ # x est la racine du generateur et n est la taille de l'echantillon
  m=2^31-1
        u=rep(0,n)
        for (i in 1:n)
        {x=(16807* x) %% m
                u[i]=x/m }
                return(u)}

generateur(3432,5)
## [1] 0.02686010 0.43768891 0.23746577 0.08725276 0.45715880
```

Remarque 2.8. *- Des méthodes plus sophistiquées, basées sur le "décalage du registre", et d'autres combinant cette dernière et congruence linéaire, permettent de générer des suites ayant une période beaucoup plus longue. Par exemple, R et Matlab 7 utilisent un générateur de type Mersenne Twister, de periode* $2^{19937} - 1$*. C'est le générateur par défaut dans le langage R (taper " ? Random" dans R pour plus d'informations).*
- Dans R, le générateur de la loi uniforme sur un intervalle est implémenté dans la fonction runif. La fonction set.seed permet de définir la racine du générateur. Fixer la racine du générateur permet notamment d'obtenir un échantillon reproductible. Comparer deux appels successifs à la fonction runif, en définissant ou non la racine :

```
set.seed(123)
runif(5)
## [1] 0.2875775 0.7883051 0.4089769 0.8830174 0.9404673
```

```
set.seed(123)
runif(5)
## [1] 0.2875775 0.7883051 0.4089769 0.8830174 0.9404673
```

Avec la même racine, on obtient le même échantillon. Par contre, si on ne fixe pas la racine, on obtient des échantillons différents à chaque appel.

```
runif(5)
## [1] 0.0455565 0.5281055 0.8924190 0.5514350 0.4566147
runif(5)
## [1] 0.9568333 0.4533342 0.6775706 0.5726334 0.1029247
```

2.2. **Simulation de lois non uniformes.**

2.2.1. **Inversion de la fonction de répartition.** Les algorithmes de simulation stochastique sont fondés sur la production de tirages de variables aléatoires distribuées selon une loi qui, bien entendu, n'est pas toujours la loi uniforme sur un intervalle. Jusqu'à présent, nous avons mis en exergue les générateurs de la loi uniforme $\mathcal{U}([0,1])$. Cela est justifié par l'argument théorique qui dit que cette loi fournit une représentation probabiliste du hasard : dans le cadre de de variables réelles, il est toujours possible de choisir l'espace probabilisé $(\Omega, \mathcal{F}, P)$ comme $([0,1], \mathcal{B}([0,1]), \mathcal{U}([0,1]))$ et donc de représenter le tirage $\omega \in \Omega$ comme le tirage d'un nombre réel dans $[0,1]$ (voir par exemple Billingsley, 1986).
Pratiquement, cela est clairement mis en valeur par le lemme suivant. Rappelons que la loi d'une v.a. X est déterminée par sa fonction de répartition $F_X(x) = P[X \leq x]$, fonction croissante de 0 à 1.

Lemme 2.9. *Soit X une v.a. à valeurs réelles de fonction de répartition F. On définit F^- l'inverse généralisée de F par*

$$F^-(u) = \inf\{x \in \mathbb{R} : F(x) \geq u\}.$$

La loi de $F^-(U)$ est la loi de X lorsque U est une v.a. de loi $\mathcal{U}([0,1])$.

Preuve : l'inverse généralisée vérifie, pour tout $u \in [0,1]$, et $x \in F^-([0,1])$,

$$F(F^-(u)) \geq u, \quad F^-(F(x)) \leq x.$$

D'où, puisque F et F^- sont croissantes, si $F^-(u) \leq x$ alors $u \leq F(F^-(u)) \leq F(x)$. De même, si $F(x) \geq u$ alors $x \geq F^-(F(x)) \geq F^-(u)$. Par conséquent,

$$\{(u,x) : F^-(u) \leq x\} = \{(u,x) : F(x) \geq u\},$$

et

$$P[F^-(U) \leq x] = P[U \leq F(x)] = F(x),$$

si U est de loi $\mathcal{U}([0,1])$. ♣

Remarque 2.10. *Lorsque la fonction de répartition est inversible, F^- coïncide avec F^{-1} et la dernière égalité est évidente. Dans la pratique, ce lemme permet de simuler des lois dont la fonction de répartition est connue explicitement et inversible.*

Exemples : proposer un algorithme pour simuler les lois suivantes à partir d'un générateur de la loi uniforme sur $[0,1]$:

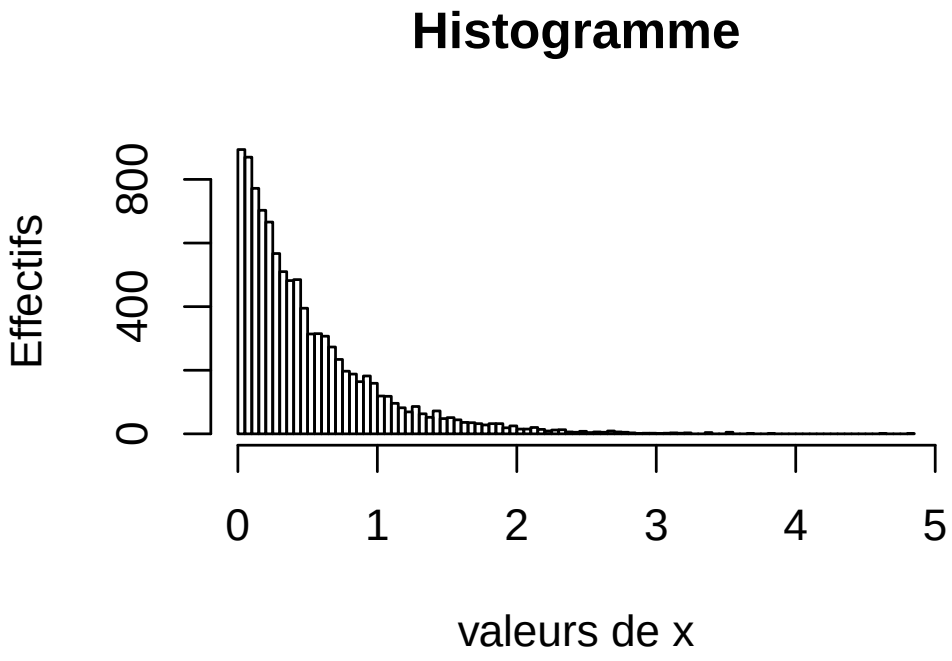
- la loi uniforme sur un intervalle $[a,b]$,

- la loi de Cauchy de densité $f(x) = \frac{1}{\pi}\frac{1}{1+x^2}$, $x \in \mathbb{R}$.

- la loi exponentielle de paramètre λ, de densité $\lambda e^{-\lambda x}\ \mathbf{1}_{\{x\geq 0\}}$.

Code R : Voici un exemple de programme R pour simuler un échantillon de taille 10000 de la loi exponentielle de paramètre $\lambda = 2$.

```
l=2
n=10000
x=-1/l *log(runif(n))
hist(x,nclass=100,main='Histogramme',ylab='Effectifs',xlab='valeurs de x')
```



Nous avons utilisé le fait que $F^{-1}(u) = -\frac{1}{\lambda}\log(1-u)$ et que $1-U$ a même loi que U si U est de loi uniforme sur $]0,1[$.

La méthode d'inversion de la fonction de répartition s'applique également pour les lois discrètes : soit X de loi discrète sur $\{x_1, x_2, \ldots, x_r\}$ donnée par $\mathbb{P}[X = x_i] = p_i$, tels que $p_i > 0$, et $\sum_{i=1}^{r} p_i = 1$.

Dans ce dernier exemple, de la loi discrète sur $\{x_1, x_2, \ldots, x_r\}$, il est facile de voir que cela est équivalent à montrer la propriété suivante :

Lemme 2.11. *Soient U une v.a. de loi uniforme sur $]0,1[$, $p_1,\dots,p_r$, tels que $p_i>0$, et $\sum_{i=1}^{r} p_i = 1$..*

Soit X la v.a. définie par $X=x_1$ si $U\in]0,p_1[$, et

$$X=x_l,\ \text{si } U\in[\ \sum_{i=1}^{l-1} p_i,\ \sum_{i=1}^{l} p_i\ [,\ \text{pour } l=2,\dots,r.$$

Alors la loi de X est donnée par $P[X=x_l]=p_l$, pour $l\in\{1,2,\dots,r\}$.

Exercice : Ecrire un algorithme pour simuler la réalisation de n lancers successifs d'une telle v.a. X.

Ce résultat se généralise facilement au cas des v.a. discrètes ayant un nombre infini dénombrable de valeurs possibles (par exemple la loi de Poisson). On pourra modifier l'algorithme de l'exercice précédent pour simuler la réalisation de n v.a. indépendantes de même loi de Poisson de paramètre λ fixé.

2.2.2. **Méthodes probabilistes.** Il existe des lois dont on ne connait pas une forme analytique exacte de la fonction de répartition, ou dont le calcul de F^- peut s'avérer difficile. On a alors recours à d'autres méthodes : une issue possible est d'utiliser des propriétés de certaines lois les liant simplement à d'autres lois simulables. L'exemple le plus connu est l'algorithme de Box et Muller (ou Algorithme polaire) pour simuler la loi normale $\mathcal{N}(0,1)$:

Lemme 2.3 : Soient U_1 et U_2 i.i.d. de même loi $\mathcal{U}([0,1])$. Si on pose

$$X_1=\sqrt{-2\log(U_1)}\cos(2\pi U_2)\ \text{ et }\ X_2=\sqrt{-2\log(U_1)}\sin(2\pi U_2),$$

alors X_1 et X_2 sont i.i.d. de loi $\mathcal{N}(0,1)$.

Preuve : soient (R,Θ) les coordonnées polaires de (X_1,X_2). Alors $R=\sqrt{{X_1}^2+{X_2}^2}=\sqrt{-2\log(U_1)}$ et $\Theta=2\pi U_2$.
On obtient ainsi que R et Θ sont indépendantes (U_1 et U_2 le sont). De plus, Θ est de loi uniforme sur $[0,2\pi]$, et la loi de R admet pour densité sur $[0,\infty[$: $f_R(r)=re^{\frac{-r^2}{2}}$. En effet, pour $r\geq 0$, $F_R(r)=P[R\leq r]=P[U_1\geq e^{\frac{-r^2}{2}}]=1-e^{\frac{-r^2}{2}}$ est dérivable de dérivée $f_R(r)$. On a alors

$$\begin{aligned}\forall x_1,x_2\in\mathbb{R},\ P[X_1\leq x_1,X_2\leq x_2] &= \int\int_{\{(r,\theta):r\cos(\theta)\leq x_1,r\sin(\theta)\leq x_2\}} f_R(r)f_\Theta(\theta)drd\theta\\ &= \frac{1}{2\pi}\int\int_{\{(r,\theta):r\cos(\theta)\leq x_1,r\sin(\theta)\leq x_2\}} e^{\frac{-r^2}{2}}rdrd\theta\\ &= \frac{1}{2\pi}\int_{-\infty}^{x_1}\int_{-\infty}^{x_2} e^{\frac{-x^2}{2}}e^{\frac{-y^2}{2}}dxdy,\ \text{ par changement de variable,}\\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x_1} e^{\frac{-x^2}{2}}dx\ .\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x_1} e^{\frac{-y^2}{2}}dy,\end{aligned}$$

ce qui démontre que les variables X_1 et X_2 sont i.i.d. de loi $\mathcal{N}(0,1)$. ♣

Exercice : Proposer un algorithme pour simuler la loi du χ^2 à k degrés de liberté. On rapelle qu'elle est définie comme étant la loi suivie par la v.a. $Y_k={X_1}^2+\ldots+{X_k}^2$, où les X_i sont i.i.d. de loi $\mathcal{N}(0,1)$.

Exercice : *Le jeu de fléchettes.* On tire sur un plan identifié à $\mathbb{R}^2$. On modélise les coordonnées X et Y du point d'impact par 2 variables gaussiennes indépendantes de même loi $\mathcal{N}(0, 25)$ (distances mesurées en cm).
On prend pour cible le disque E de centre 0 et de rayon 5 cm.

1) Quelle est la probabilité pour atteindre la cible du premier coup ?

2) On note N le nombre d'impacts sur la cible en n tirs. Sous quelle hypothèse peut-on modéliser la loi de N par une loi binomiale dont on précisera les paramètres ?

3) Quel est le plus petit rayon que l'on peut donner à la cible pour que la probabilité d'atteindre au moins une fois la cible en 10 tirs soit ≥ 0.9 ?

4) Simuler cette expérience aléatoire sur ordinateur.

Il est également utile de savoir simuler un vecteur gaussien multidimensionnel de loi quelconque. Il est possible de le faire à partir d'un vecteur gaussien centré réduit en utilisant les résultats suivants.
Dans $\mathbb{R}^n$, on notera x' le transposé d'un vecteur colonne x. De même, on notera A' la matrice transposée d'une matrice A quelconque.

Définition 2.12. *Soit $m \in \mathbb{R}^n$ et V une matrice carrée de taille $n \times n$, réelle symétrique et définie positive. On dit qu'un vecteur aléatoire X de dimension n est gaussien de loi $\mathcal{N}(m, V)$ si sa densité est définie par*

$$f(x) = \frac{1}{(2\pi)^{n/2}} \frac{1}{\sqrt{\det V}} \exp(-\frac{1}{2}(x-m)'V^{-1}(x-m)),\ x \in \mathbb{R}^n.$$

Théorème 2.13. *Un vecteur aléatoire est gaussien, de loi $\mathcal{N}(m, V)$, si et seulement si sa fonction caractéristique est de la forme*

$$\varphi_(t) = \exp(im't - \frac{1}{2}t'Vt),\ t \in \mathbb{R}^n.$$

Corollaire 2.14. *Soit X un vecteur aléatoire gaussien de loi $\mathcal{N}(m, V)$ et $Y = CX$ où C est une matrice quelconque tel que le produit CX soit défini. Alors Y est un vecteur gaussien de loi $\mathcal{N}(Cm, CVC')$.*

On en déduit une méthode simple pour simuler un vecteur gaussien X de loi $\mathcal{N}(m, V)$:
- Soit Z de loi $\mathcal{N}(0, I_n)$ dans $\mathbb{R}^n$ (I_n désigne la matrice identité dans $\mathbb{R}^n$). Le vecteur Z est facile à simuler puisque ses coordonnées sont indépendantes.
- Soit C tel que $V = CC'$. Alors le vecteur $X := m + CZ$ est de loi $\mathcal{N}(m, V)$.
Pour déterminer une telle matrice C, on peut utiliser la méthode de Choleski, basée sur le résultat suivant :

Théorème 2.15. *Soit V une matrice symétrique définie positive de taille $n \times n$. Alors il existe une matrice triangulaire inférieure L de taille $n \times n$ telle que $V = LL'$. Cette factorisation est unique si on impose aux coefficients diagonaux de L d'être strictement positifs.*

Remarque 2.16. *il est possible d'écrire un algorithme itératif (l'algorithme de Choleski) qui calcule la matrice L. Dans le logiciel R, on pourra utiliser la fonction chol.*

Exercice : nous modélisons le poids (en kg) et la taille (en cm) d'un homme adulte par deux v.a. P et T de lois respectives $\mathcal{N}(70, 100)$ et $\mathcal{N}(170, 144)$. Nous considérons que le coefficient de corrélation entre les 2 v.a. P et T est de 0.8. Simuler un échantillon aléatoire de taille 1000 du couple (P, T).

Un autre exemple de loi de probabilité est celui de la loi de Poisson, qui est liée à la loi exponentielle par le Processus de Poisson, c'est-à-dire que si N est de loi $\mathcal{P}(\lambda)$, et $(X_i)_{i\geq 0}$ sont des v.a.i.i.d. de loi $\mathcal{E}xp(\lambda)$, alors

$$P[N = k] = P[X_1 + \ldots + X_k \leq 1 < X_1 + \ldots + X_{k+1}].$$

On peut simuler une loi de Poisson $\mathcal{P}(\lambda)$ en générant des v.a. de loi $\mathcal{E}xp(\lambda)$ jusqu'à ce que leur somme dépasse 1 :

$$\begin{aligned} N &= \max\{k \geq 0 : X_1 + \ldots + X_k \leq 1\} \\ &= \max\{k \geq 0 : -\textstyle\sum_{i=1}^{k} \log(U_i) \leq \lambda\} \\ &= \max\{k \geq 0 : \textstyle\prod_{i=1}^{k} U_i \geq \exp(-\lambda)\}, \end{aligned}$$

où les U_i sont des v.a.i.i.d. de loi uniforme sur $[0, 1]$. En moyenne, combien faut-il simuler de variables de loi uniforme sur $[0, 1]$ pour obtenir une réalisation d'une v.a. de loi de Poisson $\mathcal{P}(\lambda)$? Pourquoi simuler une réalisation de la loi de Poisson est-elle coûteuse pour les grandes valeurs de λ ?

Exercice : une v.a. Y à valeurs dans $\mathbb{N}^*$ est de loi géométrique $Geo(p)$ de paramètre $p \in]0, 1[$ si pour $k \in \mathbb{N}^*$,

$$P[Y = k] = p\, q^{k-1},$$

où on a posé $q = 1 - p$.

1) Montrer que si U est une variable aléatoire de loi uniforme sur $]0, 1[$, alors

$$Z = \mathrm{Ent}\{\frac{\log(U)}{\log(q)}\} + 1$$

est une variable aléatoire de loi $Geo(p)$. Ici, $\mathrm{Ent}\{x\}$ désigne la partie entière de x.

2) En déduire une méthode pour simuler une variable aléatoire de loi géométrique $Geo(p)$.

2.2.3. **Méthode d'acceptation-rejet pour des lois admettant une densité de probabilité :** Lorsque la fonction de répartition d'une loi n'est pas inversible explicitement, et que l'on ne sait pas la relier par des propriétés probabilistes à une distribution simulable, on peut parfois avoir recours à la méthode d'acceptation-rejet. Celle-ci requiert la connaissance de la densité f de la loi, et la détermination d'une autre densité g et d'une constante $M \geq 1$ telles que $f(x) \leq Mg(x)$ sur le support de f. L'algorithme d'acceptation-rejet découle du lemme suivant :

Lemme 2.17. *La procédure*

1. *Générer x de loi de densité g, u de loi $\mathcal{U}([0, 1])$;*
2. *Si $u \leq \dfrac{f(x)}{Mg(x)}$, Alors Accepter $y := x$;*
   Sinon Retourner en **1.**

fournit une réalisation d'une v.a. de loi de densité f.

Preuve : Cet algorithme revient d'un point de vue probabiliste à considérer deux suites de v.a. indépendantes (X_i) et $(U_i), i \geq 1$. On suppose que les X_i sont i.i.d. de loi de densité g et que les U_i sont i.i.d. de loi uniforme sur $[0,1]$. On pose alors $T = \inf\{i \geq 1 : U_i \leq \frac{f(X_i)}{Mg(X_i)}\}$, et $Y = X_T$. Soient X et U indépendantes de mêmes lois que X_1 et U_1 respectivement. La fonction de répartition de Y est donnée par

$$P[Y \leq y] = \sum_{i=1}^{\infty} P[X_i \leq y; T = i]$$

$$= \sum_{i=1}^{\infty} P[X \leq y; U \leq \frac{f(X)}{Mg(X)}] \left(P[U > \frac{f(X)}{Mg(X)}] \right)^{i-1}$$

$$= \int_{-\infty}^{y} \frac{f(x)}{Mg(x)} g(x)dx \ \sum_{i=1}^{\infty} \left(\int_{-\infty}^{\infty} (\int_{\frac{f(x)}{Mg(x)}}^{1} du) g(x)dx \right)^{i-1}$$

$$= \int_{-\infty}^{y} f(x)dx \frac{1}{M} \sum_{i=0}^{\infty} (1 - \frac{1}{M})^i$$

$$\text{d'où } P[Y \leq y] = \int_{-\infty}^{y} f(x)dx. \ \clubsuit$$

Remarque 2.18. - *La variable aléatoire T est le nombre d'essais nécessaires pour accepter un nombre x. Elle est de loi géométrique*

$$P[T = k] = \frac{1}{M}(1 - \frac{1}{M})^{k-1},$$

d'espérance $E[T] = M$. Donc l'efficacité de l'algorithme est d'autant plus grande que la constante M est petite (proche de 1). Le nombre $1/M$ est appelé taux d'acceptation.

- Il est possible également de construire une méthode d'acceptation-rejet pour des variables aléatoires de loi discrète.

Exemples : Réaliser un générateur des lois dont la densité est donnée par :
- $f(x) = 3x^2 \ \mathbf{1}_{[0,1]}(x)$,

- $f(x) = (1 - |x|) \ \mathbf{1}_{[-1,1]}(x)$.

- $f(x) = \frac{1}{2} \ x^2 \ e^{-x} \ \mathbf{1}_{[0,+\infty[}(x)$, à partir de la loi exponentielle de paramètre $\lambda > 0$. Déterminer la valeur de λ maximisant le taux d'acceptation, i.e. minimisant M.

- Rappelons que la loi de Cauchy $\mathcal{C}(a)$ admet pour densité $f(x) = \frac{1}{\pi} \frac{a}{a^2 + x^2}$, $x \in \mathbb{R}$. Réaliser un générateur de la loi $\mathcal{N}(0,1)$ à partir de celui de la loi de Cauchy $\mathcal{C}(1)$. Quel est le taux d'acceptation ? Montrer que la loi de Cauchy $\mathcal{C}(1)$ est le meilleur choix de loi de Cauchy pour simuler la loi $\mathcal{N}(0,1)$.

3. Estimation par la méthode de Monte Carlo :

Une des applications les plus courantes de la simulation stochastique est l'estimation numérique d'intégrales, d'espérances de variables aléatoires ou de probabilités d'événements que l'on ne sait pas calculer de manière exacte.

3.1. **Calcul approché d'espérances par la méthode de Monte Carlo :** Soit X une variable aléatoire supposée de variance $\sigma^2 < \infty$, dont on souhaite estimer numériquement l'espérance. Considérons une suite de v.a.i.i.d. $(X_i)_{i\geq 1}$ de même loi que X. Alors par la loi forte des grands nombres,

$$\overline{X}_n := \frac{1}{n}\sum_{i=1}^{n} X_i \longrightarrow E[X], p.s.$$

et le théorème limite central nous donne la précision de cette approximation pour n grand. En effet, puisque $\dfrac{\overline{X}_n - E[X]}{\sqrt{\frac{\sigma^2}{n}}}$ converge en loi vers une v.a. U de loi normale centrée réduite, alors pour n assez grand ($n \geq 30$ en pratique convient),

$$P[E[X] \in [\overline{X}_n - t_\alpha\sqrt{\frac{\sigma^2}{n}}, \overline{X}_n + t_\alpha\sqrt{\frac{\sigma^2}{n}}]] \simeq P[U \in [-t_\alpha, t_\alpha]].$$

On choisit alors t_α telle que la probabilité $P[U \in [-t_\alpha, t_\alpha]] = 1-\alpha$ soit assez grande ($t_{0,05} = 1.96$ ou $t_{0,01} = 2.58$ en pratique). On obtient alors pour intervalle de confiance pour $E[X]$, au niveau de rejet α :

$$[\overline{X}_n - t_\alpha\sqrt{\frac{\sigma^2}{n}}, \overline{X}_n + t_\alpha\sqrt{\frac{\sigma^2}{n}}].$$

Le plus souvent, σ^2 étant inconnu, deux méthodes sont possibles :
- soit on le majore par une constante S^2.
- soit on le remplace par un estimateur de σ^2, le plus souvent l'estimateur sans biais :

$$S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2.$$

Ceci est justifié par le lemme de Slutzky.

Lemme 3.1. (Slutzky) : *Soient $(Z_n), (A_n)$ et (B_n) trois suite de v.a., une v.a. X et deux nombres réels a et b tels que X_n converge en loi vers X, A_n et B_n convergent en probabilité vers a et b respectivement lorsque n tend vers l'infini. Alors $A_nX_n + B_n$ converge en loi vers $aX + b$.*

Ce lemme implique que $\dfrac{\overline{X}_n - E[X]}{\sqrt{\frac{S_n^2}{n}}}$ converge en loi vers une v.a. de loi normale centrée réduite.

Remarque 3.2. *- La précision de cette méthode est donc d'ordre $\mathcal{O}(\frac{1}{\sqrt{n}})$. Il faut tenir compte aussi du terme σ (ou son estimateur) qui peut prendre de très grandes valeurs et rendre la méthode inutilisable en pratique. Par exemple, analyser ce problème dans le cas où on souhaiterait estimer $E[e^{\beta X}]$ par cette méthode pour X de loi normale centrée réduite et $\beta = 5$ ou 10.*
- Bien entendu, la méthode n'est utilisable que si on sait simuler la variable aléatoire X.
- Remarquons que cette méthode permet aussi d'estimer des probabilités $P[A]$ puisque $P[A] = E[\mathbf{1}_A]$. Nous avons la borne triviale $\sigma^2 = P[A](1-P[A]) \leq 1$, mais il est souvent plus précis

d'estimer σ^2 par S_n^2. Pour obtenir des réalisations des v.a. $\mathbf{1}_A$, il faut simuler l'expérience aléatoire sous-jacente et observer si l'événement A se réalise ou pas.

Exercice. Une variable aléatoire positive X est de loi de Burr $B(a,b)$ de paramètres $a>0$ et $b>0$ si elle admet pour densité de probabilité :

$$f_1(x) = ab\frac{x^{a-1}}{(b+x^a)^2}, \text{ pour } x \geq 0.$$

1) Expliquer comment simuler un échantillon de la loi de Burr $B(a,b)$ à partir d'un générateur de nombres pseudo-aléatoires de la loi uniforme sur $]0,1[$, par la méthode d'inversion de la fonction de répartition.

2) Pour quelles valeurs de a la variable aléatoire X admet-elle :
- une espérance finie ?
- une variance finie ?

3) - Écrire un algorithme permettant d'estimer l'espérance de X par la méthode de Monte Carlo, dans le cas où X est de variance finie, en le justifiant mathématiquement.
- Estimer $E[X]$ pour X v.a. de loi $B(a=3.5, b=2)$ en générant un échantillon de taille 10000, et donner un intervalle de confiance au niveau de confiance 0.95.

Exercice. Soit V une variable aléatoire positive, de loi de densité

$$f_2(x) = \frac{2}{\sqrt{\pi}}\sqrt{x}\ e^{-x}, \text{ pour } x \geq 0.$$

1) Écrire un algorithme pour simuler une réalisation de la variable aléatoire V par la méthode du rejet, à partir de la loi exponentielle de paramètre $\frac{2}{3}$ de loi de densité

$$g(x) = \frac{2}{3}e^{-2x/3}, \text{ pour } x \geq 0.$$

Au préalable, déterminer la plus petite valeur de M telle que $f_2(x) \leq Mg(x)$ pour tout $x \geq 0$.

2) Estimer $E[\cos(V)]$ en générant un échantillon de taille 10000 de variables aléatoires indépendantes de même loi que V.

Exercice. Nous modélisons les lancers simultanés de quatre dés équilibrés à 6 faces par des variables aléatoires indépendantes D_1, D_2, D_3 et D_4.
Soit l'événement $A = \{D_1 + D_2 + D_3 + D_4 \text{ est un multiple de } 3\}$.

1) Ecrire un algorithme pour estimer numériquement $P[A]$ par la méthode Monte Carlo.

2) Donner une valeur approchée de cette valeur, ainsi qu'un intervalle de confiance.

Exercice. Soit $C>0$ et $f(x) = C\left(\mathbf{1}_{[0,1]}(x) + \frac{2}{x^3}\ \mathbf{1}_{]1,+\infty[}(x)\right)$, pour $x \in \mathbb{R}$,

où $\mathbf{1}_{[a,b]}(x) = 1$, si $x \in [a,b]$, et 0 sinon.

1) Déterminer C telle que f soit une densité de probabilité.

2) Décrire deux méthodes pour simuler une variable aléatoire V de loi de densité f et écrire les algorithmes correspondants :

a) la méthode d'inversion de la fonction de répartition.

b) la méthode de rejet à partir de la loi de Cauchy de densité $h(x) = \frac{1}{\pi}\frac{1}{1+x^2}, x \in \mathbb{R}$.

Quelle est la méthode la plus efficace en temps de calcul ?

3) Ecrire un algorithme permettant d'estimer ponctuellement

$$E[\log(1+V)]$$

et d'obtenir un intervalle de confiance (au niveau de confiance 0.95), à partir d'un échantillon de taille $n = 10000$ de la v.a. V.

Exercice. Soient X, Y et Z trois variables aléatoires indépendantes de lois de Poisson de paramètres respectifs $\lambda_1 = 2$, $\lambda_2 = 3$ et $\lambda_3 = 4$.
Ecrire un algorithme calculant une estimation ponctuelle de

$$P[XYZ \geq 10],$$

ainsi qu'un intervalle de confiance au niveau de confiance 0.95, à partir d'échantillons indépendants de taille $n = 10000$ des v.a. X, Y et Z.

Indication : on pourra utiliser la fonction R *rpois(n, l)*, qui simule un échantillon de taille n de la loi de Poisson de paramètre l.

3.2. **Estimation numérique d'intégrales par la méthode de Monte Carlo :** Considérons une intégrale supposée finie

$$I = \int_D g(x)dx,$$

où $D \subset \mathbb{R}^d$ et $g : D \to \mathbb{R}$. Pour toute densité de probabilité f non nulle sur D et nulle en dehors de D, i.e.

$$f(x) \geq 0, \text{ et } \int_D f(x)dx = 1,$$

nous pouvons écrire

$$I = \int_D \frac{g(x)}{f(x)} f(x)dx = E[\frac{g(X)}{f(X)}],$$

où X est une v.a. de loi de densité f. Nous sommes donc ramener au problème de la section précédente, puisqu'il s'agit d'estimer l'espérance de la v.a. $Z = \frac{g(X)}{f(X)}$. Il suffit donc de considérer un échantillon de v.a.i.i.d. X_i de même loi que X et $Z_i = \frac{g(X_i)}{f(X_i)}, i = 1, ..., n$ et d'appliquer la méthode décrite précédemment (en supposant Z de variance finie).
Remarquons que cette méthode est de précision d'ordre $\mathcal{O}(\frac{1}{\sqrt{n}})$, indépendamment de la dimension d, ce qui rend la méthode Monte Carlo plus performante que les méthodes déterministes en grande dimension. De plus, la méthode Monte Carlo ne requiert pas d'hypothèses de régularité de la fonction g.

La précision dépend aussi de la valeur de σ^2 et donc du choix de la densité d'échantillon- -nage f. Il est judicieux de choisir f telle que l'on sache effectivement simuler X et que la variance de Z soit la plus petite possible. Différentes méthodes dites de *réduction de variance* existent. Parmi les plus connues, citons la méthode d'échantillonnage préférentiel (ou de fonction d'importance) et celle de la variable de contrôle.

- La méthode d'échantillonnage préférentiel repose sur le fait que la variance de Z est minimale pour $f(x) = \dfrac{|g(x)|}{\int_D |g(u)|du}$ (le démontrer en exercice...). Bien que l'on ne puisse pas en général choisir cette densité, car on ne sait pas calculer explicitement $\int_D |g(u)|du$, on s'en approche en choisissant une fonction h proche de $|g|$ sur D et nulle ailleurs et telle que $\int_D h(u)du$ soit calculable. On pose alors $f(x) = \dfrac{h(x)}{\int_D h(u)du}$, pour $x \in D$.

- La méthode de la variable de contrôle consiste à écrire

$$I = \int_D (g(x) - h(x))dx + \int_D h(x)dx,$$

où h est choisie proche de g sur D et telle que $\int_D h(x)dx$ soit calculable explicitement. On estime alors la première intégrale par la méthode Monte Carlo, et la variance en sera réduite. Par exemple, si $D = [a, b]$ et $|g(x) - h(x)| \leq \varepsilon$ sur D alors en échantillonnant avec la loi uniforme sur $[a, b]$, on a trivialement $\sigma \leq (b - a)\varepsilon$.

Exercice. Écrire un algorithme pour estimer numériquement par une méthode Monte Carlo les intégrales suivantes et en déterminer une valeur approchée, ainsi qu'un intervalle de confiance.

$$I_1 = \int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty \cos(x_1 + x_2 + x_3 + x_4)e^{-x_1-x_2-x_3-x_4}\, dx_1dx_2dx_3dx_4.$$

$$I_2 = \int_0^1 \int_0^1 \int_0^1 \int_0^1 e^{x_1x_2x_3x_4}\, dx_1dx_2dx_3dx_4.$$

$$I_3 = \int_0^\infty 2\tan(\sin(x))e^{-2x}dx.$$

$$I_4 = \frac{1}{\pi^2}\int_{-\infty}^\infty \int_{-\infty}^\infty \frac{\cos(x + y)}{(1 + x^2)(1 + y^2)}\, dxdy.$$

Exercice. Soit X une variable aléatoire à valeurs dans $]0, +\infty[$, de loi de densité

$$f(x) = c\ x^2 \exp(-x^3)\ \mathbf{1}_{]0,+\infty[}(x).$$

1) Déterminer la constante c.

2) Proposer un algorithme pour simuler une variable aléatoire de loi de densité f.

3) Nous souhaitons calculer une valeur approchée de l'intégrale

$$I = \int_0^{+\infty} \exp(x - x^3)dx.$$

Ecrire un algorithme pour calculer une estimation ponctuelle de I, ainsi qu'un intervalle de confiance, par la méthode Monte Carlo, en simulant un échantillon de taille $n = 10000$ de la loi de densité f.

4) Calculer à l'aide de R une estimation ponctuelle de I, ainsi qu'un intervalle de confiance.

5) Calculer une valeur approchée de l'intégrale

$$J = \int_0^1 \int_0^{+\infty} \exp(xy - x^3) dxdy,$$

en décrivant la méthode utilisée.

4. Références bibliographiques :

- C. Graham, "*Simulation stochastique et méthodes de Monte Carlo*", Editions de l'Ecole Polytechnique (2011)

- D. Knuth, "*The art of computer programming, Vol. 2* ", Third edition, Addison-Wesley (1998)

- C. Robert, "*Méthodes de Monte Carlo par chaînes de Markov*", Economica (1996)

- C. Robert, G. Casella "*Introducing Monte Carlo Methods with R*", Springer (2010)

- S. Ross, "*Simulation*", Academic Press (2002)

- B. Tuffin, "*La simulation de Monte Carlo*", Lavoisier (2010)

(Franck Vermet) EURIA, Université de Bretagne Occidentale, 6, avenue Victor Le Gorgeu, CS 93837, F-29238 BREST Cedex 3, France
E-mail address, Franck Vermet: Franck.Vermet@univ-brest.fr