

L'UTILISATION DE MÉTHODES DE SIMULATION EN ASSURANCE

PARTIE 1 : GÉNÉRER DES NOMBRES ALÉATOIRES

Frédéric PLANCHET

Laboratoire SAF¹ / JWA - Actuaire²

Julien JACQUEMIN

JWA - Actuaire³

RESUME

Cet article présente une vue d'ensemble des méthodes et outils utilisables concrètement par l'actuaire pour intégrer de manière efficace et rigoureuse des techniques de simulation dans ses modèles.

Après un rappel du contexte propre à la simulation, les générateurs de nombres aléatoires pour la loi usuelle sont présentés ; une attention particulière est apportée aux algorithmes de génération d'échantillons issus d'une loi uniforme.

Les particularités associées à la simulation des processus sont enfin brièvement abordées, ainsi que les moyens de mesurer la précision obtenue sur les résultats des simulations.

MOTS CLEFS

Simulation, méthodes de Monte-Carlo, facteurs aléatoires

1. INTRODUCTION

La mise en oeuvre de modélisations utilisant des méthodes de simulation numérique de type Monte-Carlo se développe rapidement dans le domaine de l'assurance sous l'effet conjugué :

- de l'augmentation rapide des capacités de calcul des ordinateurs,
- du besoin croissant d'évaluer le niveau de risque associé à chaque résultat produit, et donc de connaître non seulement une espérance, mais également un intervalle de confiance autour de cette espérance.

¹ ISFA, 50 avenue Tony Garnier, 69366 LYON cedex 07

² 18 avenue Félix Faure, 69007 LYON

³ 9 rue Beaujon, 75008 PARIS

De plus, l'utilisation de processus aléatoires complexes pour modéliser certaines valeurs de l'actif (actions, obligations, produits dérivés) ou du passif (garanties plancher sur des contrats en unités de compte) impose également le recours à des méthodes de simulation lorsque les formules d'évaluation fermées n'existent pas.

Enfin, il apparaît que les techniques de simulation offrent une réponse simple et efficace à la prise en compte de modèles complexes incluant des non linéarités (réassurance non proportionnelle par exemple) ou de la dépendance entre les facteurs de risque (provisionnement de rentes de retraite avec réversion).

L'objectif du présent article est de fournir une vue d'ensemble des méthodes et outils utilisables concrètement par l'actuaire pour intégrer de manière efficace et rigoureuse des techniques de simulation dans ses modèles.

Cet article est la première partie de la synthèse d'un travail de mémoire réalisé à l'ISFA en 2002 (cf. [23]).

2. RAPPEL DU CADRE GÉNÉRAL

Les méthodes de simulation s'appuient sur les théorèmes limites de la théorie des probabilités : loi des grands nombres, théorème de la limite centrale, théorème ergodique, théorème de Kolmogorov,...

L'idée sous-jacente est d'approcher le résultat théorique (qui peut être une statistique associée à une distribution, comme son espérance, sa fonction de répartition, ou, plus généralement, toute fonctionnelle associée à la distribution étudiée) en effectuant des tirages dans la loi du phénomène observé.

Pour pouvoir utiliser les modèles de simulations, que ce soit en Assurance Vie, avec les méthodes de tirages aléatoires, ou en Assurance Non-Vie, avec les modèles « fréquence-coûts », il est donc nécessaire de savoir simuler des réalisations des différentes lois de probabilité.

L'ensemble de ces techniques est généralement regroupé sous le vocable de « méthodes de Monte-Carlo ». Il en existe de différents types, adaptés à la loi que l'on souhaite simuler et au contexte.

Mais, dans tous les cas, la génération de nombres aléatoires de loi uniforme est essentielle à toute technique de simulation, du fait de l'utilisation de méthodes type « inversion de la fonction de répartition ». Autrefois présentés dans des tables de nombres aléatoires, il est désormais possible d'obtenir ces nombres grâce à l'outil informatique. Il existe deux grands types d'algorithmes générateurs de nombres aléatoires de loi uniforme : les générateurs pseudo-aléatoires, et les générateurs quasi-aléatoires.

Dans la suite on notera U une variable aléatoire réelle distribuée selon la loi uniforme continue sur l'intervalle $[0;1]$.

Cet article est organisé de la manière suivante : après une présentation générale des générateurs de nombres aléatoires pour les distributions classiques, la seconde partie s'attache à décrire les méthodes d'obtention d'échantillons issus d'une loi uniforme.

Les particularités des modèles associés à des processus régis par des équations de diffusion sont abordées succinctement dans la troisième partie.

Enfin, la quatrième partie fournit des indications sur les techniques propres à permettre d'estimer le degré de précision des calculs effectués, afin notamment de déterminer le nombre minimal de tirages nécessaire à l'obtention d'un niveau de précision préalablement fixé.

3. LES GÉNÉRATEURS DE NOMBRES ALÉATOIRES

3.1 Simulation de distributions entières et continues

Dans cette section sont présentées les méthodes classiques utilisées pour simuler des variables aléatoires issues des lois courantes dans le domaine de l'assurance, en faisant l'hypothèse que l'on est capable de générer des nombres uniformément distribués efficacement. Les techniques propres à la génération de nombres uniformément distribués sont présentées au chapitre 3.2.

3.1.1 Simulation de distributions entières

Il existe différentes méthodes de simulation de distributions entières. La plus connue est la méthode d'inversion, mais souvent cette méthode ne peut être appliquée car elle nécessite un volume de calculs trop important. C'est pourquoi nous présenterons également ici d'autres méthodes de simulation de distributions entières plus efficaces en pratique.

Soit $(p_n)_{n \geq 0}$ une distribution entière.

3.1.1.1 La méthode d'inversion

3.1.1.1.1 Présentation de la méthode

Le principe est ici de découper l'intervalle $[0;1]$ en sous intervalles dont les bornes sont les $\sum p_i$ croissantes. Ce découpage est dû au fait qu'il n'existe pas obligatoirement de bijection entre les différentes modalités de la distribution entière et l'intervalle $[0;1]$, puisque la fonction de répartition d'une distribution entière est en escaliers. L'indice de la borne supérieure de l'intervalle ainsi créé dans lequel se trouvera le nombre aléatoire de loi uniforme simulé donnera la valeur simulée de notre distribution.

Proposition⁴ : Si nous définissons la v.a.r N par

$$\left\{ \begin{array}{l} N = 0 \text{ si } U < p_0 \\ N = 1 \text{ si } p_0 \leq U < p_0 + p_1 \\ \dots \\ N = j \text{ si } \sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^j p_i \\ \dots \end{array} \right.$$

Alors N est distribuée selon la loi $(p_n)_{n \geq 0}$.

Cette méthode, très générale, peut être très coûteuse en temps de calcul dans certains cas. Pour chacune des lois entières les plus connues, il existe des méthodes de simulation particulières, qui s'avèrent bien plus performantes. Nous détaillerons quelques-unes de ces méthodes dans les paragraphes suivants.

3.1.1.1.2 Exemple trivial : la distribution de Bernoulli

La distribution de Bernoulli est la loi des variables aléatoires binaires, comme dans des scénarios de pile ou face, ou de réalisation ou non d'un événement. En ce sens, elle est extrêmement utilisée en Assurance Vie.

Définition : $(p_n)_{n \geq 0}$ est distribuée selon une loi de **Bernoulli** $B(p)$ de paramètre $p \in [0;1]$, si pour tout n entier,

$$p_n = P(N = n) = \begin{cases} q & \text{si } n = 0 \\ p & \text{si } n = 1 \\ 0 & \text{sinon} \end{cases}$$

avec $q = 1 - p$.

Le principe de la simulation d'une telle loi est donc le suivant. Comme il n'y a pas de bijection entre l'intervalle $[0;1[$ et les modalités de cette loi, nous prenons :

$$p_N^{-1}(u) = \inf \{n : p_N(n) \geq u\}, \quad 0 \leq u \leq 1.$$

Pratiquement, si une réalisation u d'une loi $U(0,1)$ est supérieure à p , alors $n = 1$.

Sinon, $n = 0$.

Exemple numérique : Soit à simuler la réalisation $(n_i)_{1 \leq i \leq l}$ d'un l -échantillon de la loi $B(0,3)$.

⁴ La démonstration de cette proposition, assez simple, est disponible dans le chapitre consacré aux simulations de l'ouvrage [1]

Comme nous le verrons dans la section concernant la simulation d'échantillons de réalisation de la loi $U(0,1)$, plusieurs algorithmes sont en concurrence.

Pour mémoire, l'espérance d'une v.a.r. suivant $B(p)$ est p . Nous allons donc comparer les espérances empiriques de simulations de I -échantillons obtenues pour différentes valeurs de I soit avec un algorithme de simulation de réalisations uniformes *Rnd* (générateur pseudo-aléatoire), soit avec un algorithme type « translation irrationnelle du Tore » (générateur quasi-aléatoire). L'espérance présentée pour le générateur *Rnd* est une moyenne sur 10 simulations.

Le générateur *Tore* est bien meilleur que *Rnd*, d'après ces résultats.

I	Min Générateur Rnd*	Max Générateur Rnd*	Moyenne empirique Générateur Rnd*	Moyenne empirique Générateur Tore
100	0,25000	0,34000	0,29800	0,30000
500	0,28800	0,33200	0,30480	0,30000
1000	0,27700	0,31400	0,29650	0,29900
2000	0,28100	0,32950	0,30160	0,29950
10000	0,29630	0,30850	0,30166	0,29990
50000	0,29646	0,30174	0,29987	0,29998

* sur 10 simulations de I réalisations

3.1.1.2 La méthode de la composition

3.1.1.2.1 Présentation de la méthode

Seconde méthode “classique” de simulation pour les distributions entières, ou même absolument continues, la méthode de la composition a pour objectif de décomposer la densité de probabilité de la variable aléatoire en mélange de fonctions de densité. Cette décomposition peut se faire à partir des probabilités conditionnelles.

Méthode : Soit X la v.a.r. dont nous cherchons une réalisation x .

La méthode de la composition propose de chercher g et Y telles que :

$$P(X = x) = \sum_i P(Y = y_i)g(x|Y = y_i).$$

Pour simuler une réalisation de X , il faut alors simuler un premier nombre aléatoire u_1 pour trouver y réalisation de Y telle que $P(Y = y) = u_1$ puis générer un autre nombre aléatoire u_2 afin d'en déduire $x = g(u_2|Y = y)$.

3.1.1.2.2 Exemple : Simulation d'un échantillon Binomial Négatif

Définition : $(p_n)_{n \geq 0}$ est distribuée selon une loi **Binomiale Négative** $BN(r,p)$ de paramètres $r > 0$ et $p \in [0;1]$, si pour tout n entier,

$$p_n = P(N = n) = \frac{\Gamma(r+n)}{\Gamma(r)n!} p^r q^n,$$

avec $q = 1 - p$, et

$$\frac{\Gamma(r+n)}{\Gamma(r)} = \begin{cases} (r+n-1)\dots r & \text{si } n \geq 1 \\ 1 & \text{si } n = 0. \end{cases}$$

La loi Binomiale Négative est en réalité une distribution Poisson-mélange. Cette propriété étant à la base de la méthode de simulation de la distribution présentée, nous allons tout d'abord rappeler les définitions nécessaires à sa bonne compréhension.

Définition : Distributions Poisson-mélange

Soit un couple (Λ, N) de v.a.r., avec $\Lambda > 0$, pour lequel sont connus :

La distribution conditionnelle de N/Λ : loi de Poisson

La distribution marginale de Λ , une fonction de répartition F .

La loi marginale de N a pour expression :

$$p_n = p_n(F) = P(N = n) = \int_0^{+\infty} P(N = n / \Lambda = \lambda) dF(\lambda) = \int_0^{+\infty} e^{-\lambda} \frac{\lambda^n}{n!} dF(\lambda)$$

avec $p_n > 0$ pour tout $n \in N$.

La distribution entière ainsi obtenue est une distribution Poisson F -mélange.

Propriété : Une distribution Binomiale Négative $BN(r, p)$ est une distribution Poisson-Gamma $(r, \frac{p}{q})$.

Ainsi, il est possible d'utiliser cette propriété pour simuler une distribution Binomiale Négative, à partir de la méthode de la composition, en deux étapes :

- ✓ simulation d'une réalisation λ de la loi Gamma $(r, p/q)$,
- ✓ simulation d'une réalisation de la loi de Poisson de paramètre λ . Nous obtenons ainsi une réalisation de la loi $BN(r, p)$.

Remarque : Dans cet exemple, et contrairement à la présentation de la méthode, c'est la loi discrète (loi de Poisson) à laquelle nous appliquons le conditionnement, et non pas la loi absolument continue, mais il est bien évident que le principe reste le même.

3.1.1.2.3 Généralisation aux distributions Poisson-mélange

Si nous supposons la loi Λ définie précédemment comme absolument continue, alors la méthode proposée ci avant pour la distribution Binomiale Négative (Poisson-Gamma) se généralise facilement.

Ainsi, il est possible de simuler une distribution Poisson F -mélange en deux étapes :

- ✓ simulation d'une réalisation λ de la loi Λ ,
- ✓ simulation d'une réalisation de la loi de Poisson de paramètre λ , qui permet d'obtenir une réalisation de la loi Poisson F -mélange.

Pour conclure cette section, nous noterons que certaines distributions entières ne sont pas simulées à partir des deux méthodes présentées ici, notamment la loi de Poisson, essentielle dans le milieu de l'Assurance non-vie, et la loi Binomiale. Les algorithmes correspondants sont présentés ci-après.

3.1.1.2.4 Cas de la distribution de Poisson et de la loi binomiale

- ✓ Distribution de Poisson $P(\lambda)$, de paramètre $\lambda > 0$.

Loi fondamentale en assurance non-vie, la loi de Poisson est simulée le plus souvent à partir d'un algorithme différent de celui proposé par la méthode d'inversion. Rappelons tout d'abord la définition d'une loi de Poisson.

Définition : $(p_n)_{n \geq 0}$ est distribuée selon une loi de **Poisson** $P(\lambda)$ de paramètre $\lambda > 0$, si pour tout $n \geq 0$, n entier,

$$p_n = P(N = n) = e^{-\lambda} \frac{\lambda^n}{n!}.$$

Cette méthode se base sur une propriété liant loi de Poisson et loi Exponentielle (nous verrons au paragraphe 3.1.2.1.2 comment simuler une telle loi).

Propriété : Soit V_1, V_2, \dots une suite de v.a.r. i.i.d. suivant une loi Exponentielle de paramètre 1.

Si N est le plus petit entier k positif tel que :

$$\sum_{i=1}^{k+1} V_i > \lambda,$$

alors N suit une loi de Poisson.

Cette méthode est très simple à mettre en œuvre, dès lors qu'un algorithme de simulation de réalisations d'une loi exponentielle est connu.

- ✓ Distribution Binomiale $B(a,p)$

Définition : $(p_n)_{n \geq 0}$ est distribuée selon une loi **Binomiale** $B(a,p)$ de paramètres $a \in \mathbb{N}^*$ et $p \in [0;1]$, si pour tout n entier avec $n \in \{0, \dots, a\}$,

$$p_n = P(N = n) = \binom{a}{n} p^n q^{a-n},$$

avec $q = 1 - p$.

Propriété : Semi-additivité

Soient deux v.a.r. indépendantes B_1 et B_2 , B_1 suivant $B(a_1, p)$ et B_2 suivant $B(a_2, p)$.

Alors la v.a.r. $B = B_1 + B_2$ est distribuée selon la loi $B(a_1 + a_2, p)$.

La méthode de simulation découle directement de cette propriété. En effet, comme nous l'avons vu précédemment, il est très simple de simuler des v.a.r indépendantes et distribuées selon une loi de Bernoulli, qui n'est autre qu'une loi binomiale de paramètres $a = 1$ et p . Or, comme a est ici un entier, il suffit de sommer les résultats de a simulations de loi Bernoulli de paramètre p , pour obtenir une réalisation d'une loi $B(a, p)$. Pour obtenir un n - échantillon i.i.d. d'une telle loi Binomiale, il suffit de recommencer l'opération n fois, de façon indépendante.

3.1.2 Simulation de distributions continues

Après avoir étudié les méthodes de simulation des distributions entières, utilisées en général pour modéliser des fréquences de sinistres, il est désormais essentiel de voir comment simuler des distributions continues, qui permettront de modéliser des coûts de sinistres ou des taux d'intérêts, par exemple.

Tout au long de cette partie, nous mettrons en œuvre des exemples basés sur la simulation de la loi Normale que nous départagerons par l'application de tests statistiques identiques. Outre ces exemples "normaux", nous présentons dans cette section les algorithmes de simulation des lois continues les plus usitées, le but de cette étude étant, entre autres, de fournir un support de programmation à tout actuaire voulant utiliser des méthodes stochastiques en tarification ou provisionnement.

3.1.2.1 Méthode de l'inversion de la fonction de répartition**3.1.2.1.1 Présentation de la méthode**

Classiquement, c'est l'une des méthodes les plus utilisées en simulation, au moins lorsque la puissance de l'outil informatique permet les calculs, et que l'inversion de la fonction de répartition est possible, ce qui nécessite le plus souvent une expression analytique simple de cette fonction.

Propriété : Si la fonction de répartition F est continue et strictement croissante sur $[0;1]$, alors elle admet une fonction réciproque notée F^{-1} .

L'existence de la fonction de répartition inverse (ou réciproque) étant acquise, son utilisation est légitimée par le fait que si U est une v.a.r. suivant la loi uniforme sur l'intervalle $[0;1]$ alors la v.a.r. $F^{-1}(U)$ a pour fonction de répartition F .

Ainsi, pour simuler un n -échantillon i.i.d. d'une loi ayant pour fonction de répartition F , il suffit de simuler n réalisations indépendantes d'une v.a.r. de loi uniforme sur l'intervalle $[0;1]$, puis d'appliquer l'inverse de la fonction de répartition à chacune de ces valeurs. Ceci montre tout l'enjeu d'une simulation "optimale" des réalisations d'une loi uniforme, puisque les seules approximations résident dans cette opération quand l'inversion de la fonction de répartition est possible de manière analytique.

3.1.2.1.2 Exemples

➤ Loi Exponentielle $E(\theta)$

Cette loi dépend d'un unique paramètre $\theta > 0$.

Définition : Fonction de répartition de la loi Exponentielle

La fonction de répartition F de la loi exponentielle est donnée par :

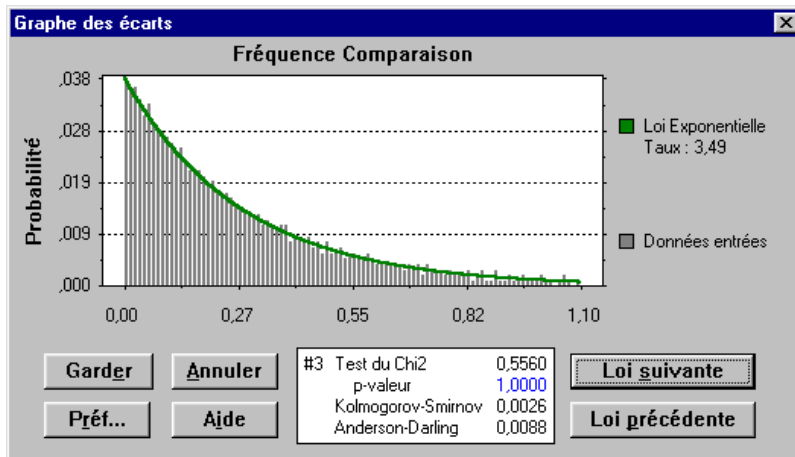
$$F(x) = 1 - e^{-\theta x}.$$

Grâce à l'expression analytique de cette fonction de répartition, nous pouvons facilement extraire :

$$F^{-1}(u) = -\frac{1}{\theta} \ln(1-u).$$

L'essentiel de la programmation réside alors dans le choix de l'algorithme de simulation de nombres aléatoires de loi uniforme sur $[0;1]$.

Exemple numérique : Soit à simuler la réalisation $(x_i)_{1 \leq i \leq 1000}$ d'un 1000-échantillon de la loi $E(3,5)$. Pour cet exemple, et comme dans la suite de cette étude, nous choisissons ici l'algorithme de translation irrationnelle du Tore pour obtenir les nombres aléatoires de loi uniforme. Le graphique suivant présente les résultats obtenus :



Comme le montre ce graphique, et également la P-valeur du test du Chi-Deux, cet échantillon de 1000 valeurs est bien distribué selon une loi exponentielle de paramètre $\theta = 3,5$, les seuls défauts de la simulation résidant dans la qualité des valeurs de loi uniforme générées.

➤ **Loi Normale $N(\mu; \sigma^2)$: Algorithme de De Moro**

Comme nous le verrons dans la suite de cette étude, il existe de très nombreux algorithmes de simulation de la loi Normale. Après avoir présenté la fonction de répartition de cette loi, nous développerons ici l'un des algorithmes les plus performants de simulation existant pour obtenir des échantillons gaussiens.

Les paramètres de la loi Normale sont μ réel, et $\sigma > 0$.

Définition : Fonction de répartition de la loi Normale

La fonction de répartition F de la loi Normale est donnée par :

$$F(x) = \Phi\left(\frac{x - \mu}{\sigma}\right)$$

où Φ est la fonction de répartition de la loi Normale centrée réduite, $N(0;1)$,

avec :

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt .$$

Le problème de la simulation de $N(\mu; \sigma^2)$ peut donc être vu comme le problème de l'inversion de la fonction de répartition de $N(0,1)$. En effet, si $Y \sim N(0;1)$, alors $X = \mu + \sigma Y$ suit $N(\mu; \sigma^2)$. L'inverse de la fonction de répartition de la loi Normale centrée réduite n'étant pas aisément calculable, l'algorithme de De Moro permet d'approcher le résultat par une méthode numérique. Il est à noter que cette méthode de simulation est l'une des plus précises parmi l'ensemble des algorithmes existants, notamment au niveau de la queue de distribution.

Définition : Algorithme de De Moro

Soit y la valeur de loi uniforme générée, alors $y = \Phi(x)$ d'après ce qui précède.

Posons $z = y - 0,5$.

✓ Si $|z| \leq 0,42$ alors x est approché par :

$$x = z \frac{\sum_{i=0}^3 a_i z^{2i}}{\sum_{j=0}^4 b_j z^{2j}} .$$

✓ Dans le cas contraire, si $|z| > 0,42$, alors :

$$x = \varepsilon \left(\sum_{i=0}^8 c_i T_i(t) \right) - \varepsilon \frac{c_0}{2}, \text{ avec } \varepsilon \text{ signe de } z, \text{ et}$$

$$t = k_1 \left\{ 2 \ln \left(-\ln \left(\frac{1}{2} - |z| \right) \right) - k_2 \right\},$$

la fonction $f(t) = \left(\sum_{i=0}^8 c_i T_i(t) \right) - \frac{c_0}{2}$ pouvant être approchée par l'algorithme

suivant :

Soient $d_{10} = 0$ et $d_9 = 0$.

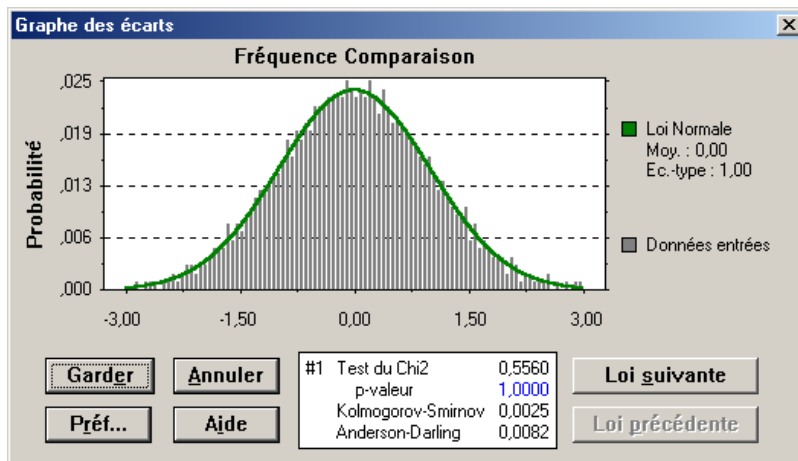
Soient d_i les réels déterminés par la relation récursive :

$$d_i = 2td_{i+1} - d_{i+2} + c_i \text{ pour } i=8,7,\dots,1.$$

$$\text{Alors } f(t) = td_1 - d_2 + \frac{c_0}{2}.$$

Les valeurs a_i, b_i, c_i, k_i sont des paramètres réels disponibles en annexe au présent article.

Exemple numérique : Soit à simuler la réalisation $(x_i)_{1 \leq i \leq 1000}$ d'un 1000-échantillon de la loi $N(0,1)$. En utilisant toujours le même algorithme de simulation de v.a.r. de loi uniforme, on obtient le graphique suivant :



Nous remarquons encore ici la qualité de la simulation effectuée, selon la P-valeur du test du Chi-deux, égale à 1.

3.1.2.2 Méthode du rejet

3.1.2.2.1 Présentation de la méthode

Cette méthode est basée sur le résultat suivant :

Lemme : Soit X une variable aléatoire de densité $f_X(x)$.

Soient $C \geq 1$ une constante, h une fonction de densité, et g une fonction réelle à valeurs dans l'intervalle $]0;1[$, définies telles que :

$$f_X(x) = Ch(x)g(x)$$

Soient alors U et V deux v.a.r, U de loi uniforme sur $[0;1]$, V de densité h .

Alors la loi conditionnelle de V sachant $U \leq g(V)$ est de densité f_X , c'est à dire :

$$f_V(x | U \leq g(V)) = Ch(x)g(x) = f_X(X)$$

Preuve : Par application de la formule de Bayes à la densité de la loi conditionnelle de V sachant $U \leq g(V)$,

$$f_V(x | U \leq g(V)) = \frac{P(U \leq g(V) | V = x)h(x)}{P(U \leq g(V))}$$

de plus, $P(U \leq g(v) | V = x) = P(U \leq g(x)) = g(x)$

$$\text{et } P(U \leq g(V)) = \int_{-\infty}^{+\infty} P(U \leq g(V) | V = x)h(x)dx = \int_{-\infty}^{+\infty} P(U \leq g(x))h(x)dx$$

$$= \int_{-\infty}^{+\infty} g(x)h(x)dx = \frac{1}{C} \int_{-\infty}^{+\infty} f_X(x)dx = \frac{1}{C} \text{ par définition de } C, h, g,$$

et par propriété de f_X .

Au final, il vient : $f_V(x | U \leq g(V)) = Cg(x)h(x) = f_X(x)$.

Remarque : Le choix de h , g et C doit être fait comme suit :

- ✓ h est une fonction de densité "simple", que nous savons simuler,
- ✓ C est une constante le plus proche de 1 que possible (ce qui améliore la qualité du générateur),
- ✓ g en découle, il faut prendre $g(x) = \frac{f_X(x)}{Ch(x)}$ sur l'intervalle des x tels que $f_X(x) > 0$.

Algorithme : La méthode du rejet, découlant de ce lemme, est donc utilisée comme suit :

Nous simulons un couple (U_1, V_1) de v.a.r. indépendantes de lois respectives $U([0;1])$ et la loi de densité h .

Si $U_1 \leq g(V_1)$, alors nous gardons $X = V_1$.

Si $U_1 > g(V_1)$, alors nous rejetons V_1 , et nous générons indépendamment un nouveau couple (U_2, V_2) , sur lequel nous recommençons les mêmes tests.

Remarque : Il est à noter que le lemme est parfois énoncé sous une forme voisine, proposant de trouver une fonction $Ch(x)$, avec C et h définis comme précédemment, qui majore $f_X(x)$, puis d'effectuer des tests avec la même fonction g que celle que nous avons définie.

3.1.2.2.2 Exemple

Loi Normale $N(0;1)$: Méthode du rejet polaire⁵

Cette méthode permet de simuler des n -échantillons de couples de variables aléatoires normalement distribuées.

Définition : Fonction de densité de la loi Normale

La fonction de densité f de la loi Normale $N(\mu, \sigma^2)$ est donnée par :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\left(\frac{x-\mu}{\sigma}\right)^2}{2}}.$$

En particulier, la fonction de densité de la loi Normale centrée réduite $N(0;1)$ est :

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x)^2}{2}}.$$

*Méthode du rejet polaire*⁶ : Soient deux variables aléatoires indépendantes u_1 et u_2 distribuées selon la loi uniforme sur l'intervalle $[-1;1]$.

A chaque couple (u_1, u_2) simulé, nous testons s'il se trouve dans le cercle trigonométrique, c'est à dire si $u_1^2 + u_2^2 \leq 1$.

Ainsi la densité de probabilité jointe est :

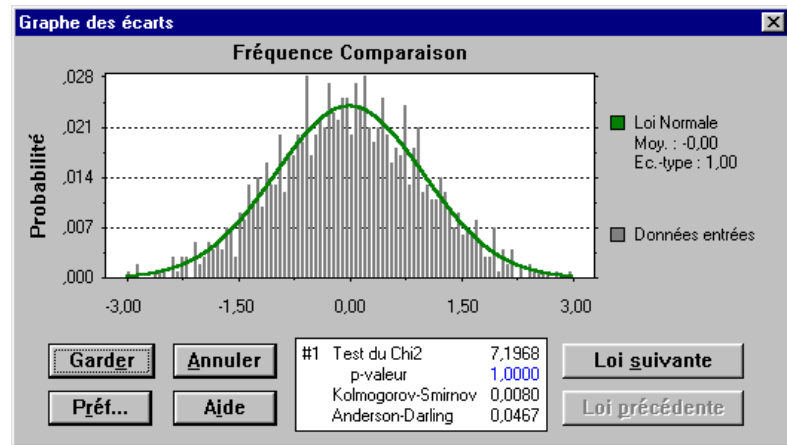
$$f_{u_1, u_2}(u_1, u_2) = \frac{1}{\pi}.$$

Alors, $y_1 = Ku_1$ et $y_2 = Ku_2$ sont des v.a.r. normalement distribuées et de corrélation nulle, avec : $K = \sqrt{\frac{-2 \ln t}{t}}$ et $t = u_1^2 + u_2^2$.

Exemple numérique : Soit à simuler la réalisation $(x_i)_{1 \leq i \leq 1000}$ d'un 1000-échantillon de la loi $N(0;1)$. En utilisant toujours le même algorithme de simulation de v.a.r. de loi uniforme, on obtient le graphique suivant :

⁵ Méthode présentée pour la première fois dans l'article [8] en 1964

⁶ Pour la preuve, voir notamment [8]



En comparaison avec les résultats obtenus avec l'algorithme d'inversion de la fonction de répartition de De Moro, cet algorithme s'avère moins précis en tout cas pour une taille d'échantillon simulé assez faible, bien que la P-valeur du test du Chi-deux reste égale à 1,0000.

3.1.2.3 Méthode de la composition

3.1.2.3.1 Présentation de la méthode

Utilisée pour les simulations de distributions entières, la méthode de la composition se généralise aux distributions absolument continues. Pour mémoire, cette méthode a pour objectif de décomposer la densité de probabilité de la variable aléatoire en mélange de fonctions de densité. Cette décomposition peut se faire à partir des probabilités conditionnelles.

Méthode : Soit X la v.a.r. dont nous cherchons une réalisation x .

Soit f_x la densité de probabilité de X .

La méthode de la composition se propose de chercher g et F_Y telles que :

$$f_X(x) = \int g(x|y) dF_Y(y).$$

Pour simuler une réalisation de X , il faut alors simuler un premier nombre aléatoire u_1 pour trouver y réalisation de Y telle que $F_Y = u_1$ puis de générer un autre nombre aléatoire u_2 afin d'en déduire $x = g(u_2|Y = y)$.

Enfin, il faut remarquer que certaines lois ne sont pas simulées à partir des trois méthodes précédentes.

3.2 Génération de nombres aléatoires de loi uniforme : générateurs pseudo-aléatoires et quasi-aléatoires; notion de suite à discrédance faible.

La partie précédente a montré l'importance de la génération de nombres aléatoires de loi uniforme dans les méthodes de simulation. En effet, l'optimalité de la plupart des algorithmes présentés réside presque uniquement dans la qualité de la répartition des échantillons distribués selon une loi uniforme sur l'intervalle de définition. C'est pourquoi nous présentons ici plusieurs types d'algorithmes de génération de loi uniforme dans le présent paragraphe.

3.2.1 Les générateurs pseudo-aléatoires

Ces générateurs sont classiquement les plus connus. Ils sont dits pseudo-aléatoires dans le sens où la suite des nombres qui sera simulée est prévisible, sous certaines conditions.

3.2.1.1 Les générateurs congruentiels

Basés sur l'algorithme inventé par D.H. Lehmer en 1948, les générateurs congruentiels font partie de la famille des générateurs pseudo-aléatoires. La particularité de ce type de générateurs est qu'ils sont initialisables : ils sont basés sur une « graine », qui est la première valeur de la suite générée, et ensuite sur une formule récursive de calcul des différentes valeurs de la suite. Changer de graine permet ainsi d'obtenir une suite différente. Le choix de cette graine est d'ailleurs essentiel à la qualité de la simulation, comme on le verra par la suite. D'un point de vue pratique, la présence d'une graine est très intéressante, car elle permet de trouver les mêmes résultats lors de plusieurs simulations identiques (et la graine devient un paramètre de l'étude). Par ailleurs il est important de noter ici que les principaux langages informatiques intègrent un générateur congruentiel dans leur bibliothèque interne de fonctions.

3.2.1.1.1 Générateur congruentiel multiplicatif

Ce générateur est le plus simple des générateurs congruentiels existants.

Définition : Le **générateur multiplicatif** produit une suite $(X_n)_{n \geq 0}$ d'entiers définie par :

- ✓ le choix d'une graine : $X_0 \in \mathbb{N}^*$,
- ✓ le calcul récursif : $X_{n+1} = (k \times X_n) \bmod m$,

où k et m sont des entiers positifs, et \bmod l'opérateur « reste de la division euclidienne ».

Il suffit ensuite de diviser par m les X_i obtenus ainsi pour obtenir les réalisations u_i d'une var U de loi uniforme sur l'intervalle $[0,1]$: $u_i = X_i / m$.

3.2.1.1.2 Cas général des générateurs congruents

3.2.1.1.2.1 Méthode

Définition : Un **générateur congruentiel** est défini par :

- ✓ le choix d'une graine : $X_0 \in N^*$,
- ✓ le calcul récursif : $X_{n+1} = (k \times X_n + p) \bmod m$,

où k , m et p sont des entiers positifs, et \bmod l'opérateur « reste de la division euclidienne ».

3.2.1.1.2.2 Remarques

Tout d'abord, il faut noter qu'il y a au mieux m réalisations générées, avec une équiprobabilité $1/m$. La valeur 0 peut de plus « sortir » au cours d'un tirage ce qui peut poser des problèmes d'erreurs informatiques, comme des divisions par 0. De plus, ces générateurs produisent des valeurs qui peuvent s'approcher de la valeur 1, mais sans jamais l'atteindre, l'intervalle des valeurs obtenues étant en réalité l'intervalle semi-ouvert $[0;1[$.

D'autre part, le choix des valeurs des différents paramètres est essentiel à l'optimisation du générateur. Si p est choisi non nul, la période du générateur peut être complète, c'est à dire égale à m , si les trois critères suivants sont respectés :

*Proposition*⁷ : Dans le cas où p est non nul, la période du générateur congruentiel correspondant est complète si :

- ✓ p et m n'ont aucun diviseur commun,
- ✓ $k = 1 \bmod h$ où h est un facteur premier de m ,
- ✓ $k = 1 \bmod 4$ si m est un multiple de 4.

Exemple : En choisissant par exemple le générateur $X_{n+1} = (25 \times X_n + 16) \bmod 256$, c'est à dire en négligeant le premier critère de la propriété, le choix de $X_0 = 12$ donne la série suivante :

i	X _i
0	12
1	60
2	236
3	28
4	204
5	252

Tous les entiers produits sont pairs !

De même, avec $X_0 = 11$, tous les entiers produits sont impairs.

Enfin, au niveau des propriétés essentielles de ces générateurs congruents, Greenberger⁸ a démontré en 1961 la propriété suivante :

⁷ Pour la preuve voir [2]

⁸ Dans l'article [12]

Propriété : Le coefficient de corrélation entre les variables X_i et X_{i+1} est compris dans l'intervalle :

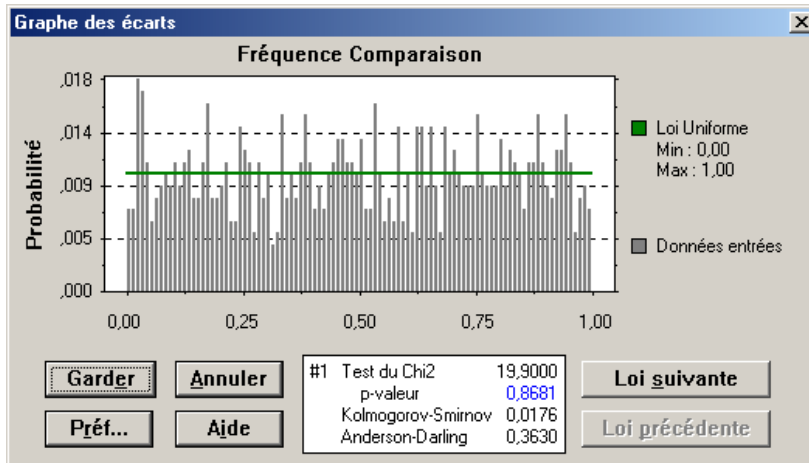
$$\left[\frac{1}{k} - \left(\frac{6p}{km} \right) \left(1 - \frac{p}{m} \right) - \frac{k}{m}; \frac{1}{k} - \left(\frac{6p}{km} \right) \left(1 - \frac{p}{m} \right) + \frac{k}{m} \right].$$

Remarque : Les générateurs en langage binaire, comme par exemple le générateur *Rnd* implémenté dans les bibliothèques de macros du logiciel *Microsoft Excel*, choisissent en général $m = 2^a$ avec a entier positif correspondant au nombre de bits-1. La variable k est telle que $k = 8r \pm 3$ avec r entier positif choisi pour avoir une période optimale pour la suite.

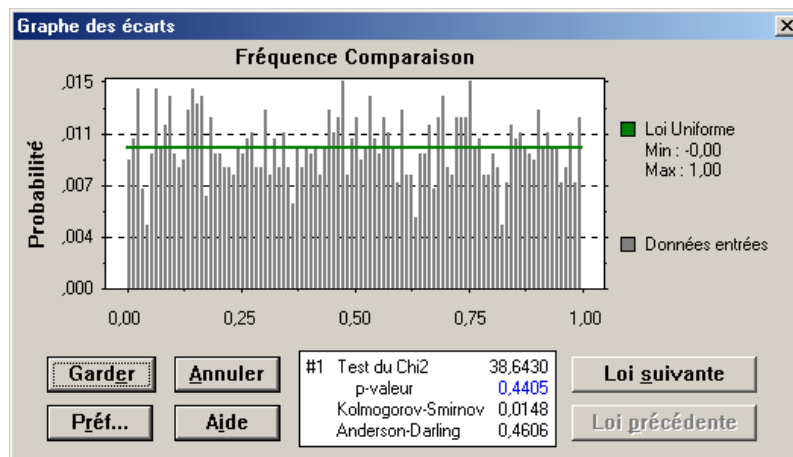
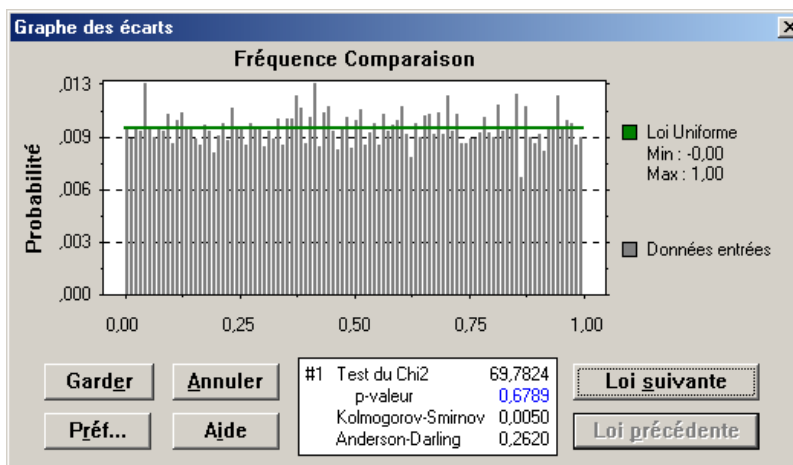
3.2.1.1.2.3 Applications

A partir du générateur congruentiel *Rnd* de *Visual Basic*⁹ (langage de développement *Microsoft*), nous avons effectué quelques tests statistiques d'adéquation à la loi uniforme. Les tests ont été effectués sur des simulations de 1 000, 2 000, puis 10 000 valeurs. Une lecture graphique de ces simulations permet de prendre un peu de recul face à ce type de générateurs :

✓ Simulation *Rnd* avec 1 000 valeurs



⁹ Identique à la fonction *Alea* de *Microsoft Excel*

✓ Simulation *Rnd* avec 2 000 valeurs✓ Simulation *Rnd* avec 10 000 valeurs

Même si ce type de générateurs semble pouvoir “tromper” les tests statistiques les moins exigeants comme le test du Chi-Deux, il est tout de même à noter qu’en dessous de 10 000 valeurs, les séquences générées ont graphiquement tendance à former des paquets de nombres sur certaines parties de l’intervalle $[0;1]$. Ceci peut avoir pour conséquence en pratique de biaiser significativement les simulations utilisant ce type d’algorithme.

3.2.1.2. Autres générateurs pseudos-aléatoires

Les générateurs congruentiels nécessitant des multiplications, il est possible de trouver dans la littérature d'autres générateurs pseudos aléatoires, n'utilisant pas cette opération coûteuse en temps de calcul. Nous les citerons ici très brièvement :

- ✓ Le générateur de Mitchell et Moore

$$X_n = (X_{n-24} + X_{n-55}) \text{ mod } m.$$

- ✓ Le générateur de Knuth

$$X_n = (X_{n-24} - X_{n-55}) \text{ mod } m.$$

Comme l'a montré l'exemple de la fonction *Rnd*, ces générateurs ne sont pas, en général, optimum quant à la répartition des réalisations sur l'intervalle [0;1]. C'est pourquoi nous allons étudier un autre type de générateurs de nombres aléatoires distribués selon une loi uniforme, les générateurs quasi-aléatoires.

3.2.2 Les générateurs quasi-aléatoires

Ces générateurs se basent sur la construction de suites à discrédance faible. Rappelons d'abord la définition de la discrédance¹⁰.

Définition : Discrédance locale

Soient : $x = (x_n)_{n \geq 1}$ une suite de points de $[0;1]^n$,

λ_n la mesure de Lebesgue sur $[0;1]^n$,

A un sous pavé quelconque de $[0;1]^n$.

La discrédance locale d'ordre k de x par rapport à A est la quantité :

$$D_k(A,x) = \frac{1}{k} \text{Card} \{ i \in \{1, \dots, k\} \text{ tq } x_i \in A \} - \lambda_n(A).$$

Définition : Discrédance

Soient : $x = (x_n)_{n \geq 1}$ une suite de points de $[0;1]^n$,

λ_n la mesure de Lebesgue sur $[0;1]^n$,

A un sous pavé quelconque de $[0;1]^n$,

P l'ensemble des sous pavés de $[0;1]^n$.

La discrédance d'ordre k de la suite x est la quantité :

$$D_k^\infty(x) = \text{Sup} \{ | D_k(A,x) | \text{ tq } A \in P \}.$$

¹⁰ Tous les détails sont très bien développés dans [19]

Intuitivement, ces discrédances mesurent la qualité de la répartition des valeurs de la suite x sur son intervalle de définition. Elle est d'autant plus faible que x est bien répartie sur $[0,1]^n$. Ces définitions permettent alors de définir les suites équiréparties, puis les suites à discrédance faible.

Définition : Suites équiréparties

Soit $x = (x_n)_{n \geq 1}$ une suite de points sur $[0,1]^n$, et P tel que défini ci-dessus.

La suite x est dite équirépartie, ou uniformément distribuée, si et seulement si pour tout pavé A de P , la limite quand $k \rightarrow +\infty$ de la discrédance locale d'ordre k selon A est égale à 0.

Les suites pseudo-aléatoires susnommées sont équiréparties¹¹.

Définition : Suite à discrédance faible

Soit $x = (x_n)_{n \geq 1}$ une suite de points sur $[0,1]^n$.

La suite x est à discrédance faible si :

$$D_k^\infty(x) = O_{k \rightarrow \infty} \left(\frac{(\ln k)^n}{k} \right).$$

La discrédance des suites à discrédance faible est bien meilleure que celle des suites pseudo-aléatoires¹². Ainsi, les algorithmes décrits ci-dessous produisent des suites qui sont asymptotiquement meilleures que les suites de nombres pseudo-aléatoires. Ils remplacent les algorithmes pseudo-aléatoires utilisés traditionnellement dans les simulations de Monte-Carlo, et sont ainsi à la base des simulations de Quasi-Monte-Carlo. Il est à noter que ce type d'algorithme est particulièrement utilisé pour le calcul numérique d'intégrales, notamment pour sa rapidité de convergence.

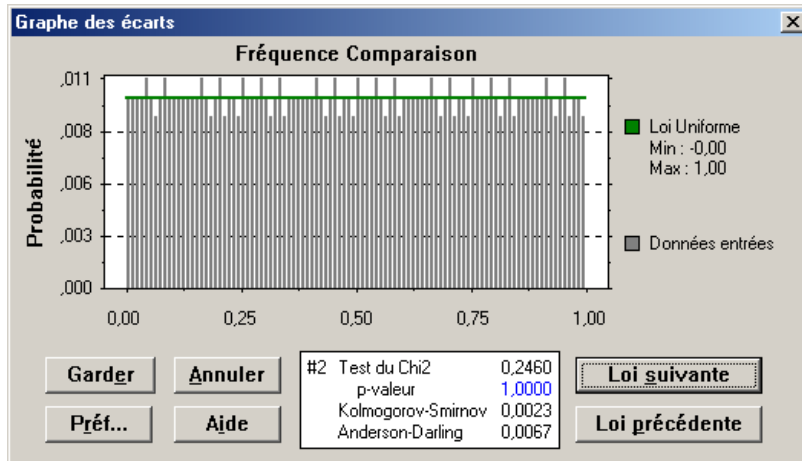
Les suites quasi-aléatoires les plus connues, et les plus simples, sont les suites de Van der Corput. Elles permettent de générer des suites de nombres distribués selon une loi uniforme. Elles sont basées sur la conversion d'entiers dans la base d'un nombre premier (au choix), puis inversion par rapport à la virgule décimale.

Un simple test graphique sur 1 000 valeurs montre que ces suites génèrent des échantillons mieux répartis que ceux obtenus avec la fonction *Rnd*.

¹¹ Voir éventuellement la preuve dans [19]

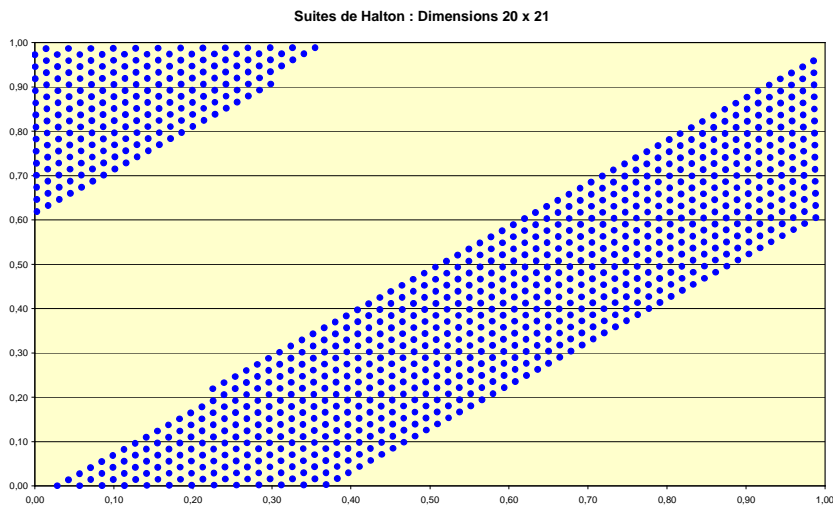
¹² Cette proposition fait l'objet d'un développement très intéressant dans [19]

✓ Simulation Van der Corput de base 2 avec 1000 valeurs



Ces suites de Van der Corput sont à la base de méthodes quasi-aléatoires de génération d'échantillons multiuniformes : les suites de Halton, les suites de Fauré et les suites de Sobol, dans l'ordre croissant de qualité, mais aussi de complexité. Malgré leur efficacité, ces suites montrent des corrélations importantes entre les suites générées, à partir d'un nombre élevé de dimensions.

Exemple : Corrélations des 20^{ème} et 21^{ème} suites de Halton tracées dans le plan



C'est pourquoi les utilisateurs de telles suites se limitent en pratique à 6 ou 8 dimensions.

La méthode de la Translation Irrationnelle du Tore est une autre méthode de génération quasi-aléatoire d'échantillons de loi multiuniforme. Cette méthode a été longuement présentée dans [19], est beaucoup plus simple à implémenter que les précédentes, et s'avère particulièrement efficace. Nous ne reprendrons ici que la présentation de l'algorithme, en nous attardant brièvement sur le fait qu'elle est d'une grande simplicité informatique, et que les résultats obtenus sont très satisfaisants.

Définition : Algorithme d'implémentation des suites de translation irrationnelle du Tore (dimension s)

La suite implémentée est :

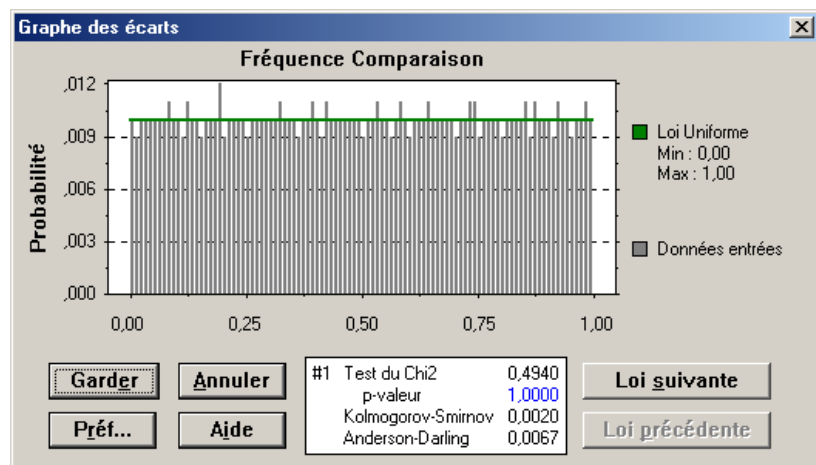
$$x_n = (\{n\sqrt{p_1}\}, \dots, \{n\sqrt{p_s}\})$$

où (p_1, \dots, p_s) représente le vecteur des s premiers nombres premiers,

et $\{ \}$ représente l'opérateur *partie fractionnaire* (le nombre moins sa partie entière).

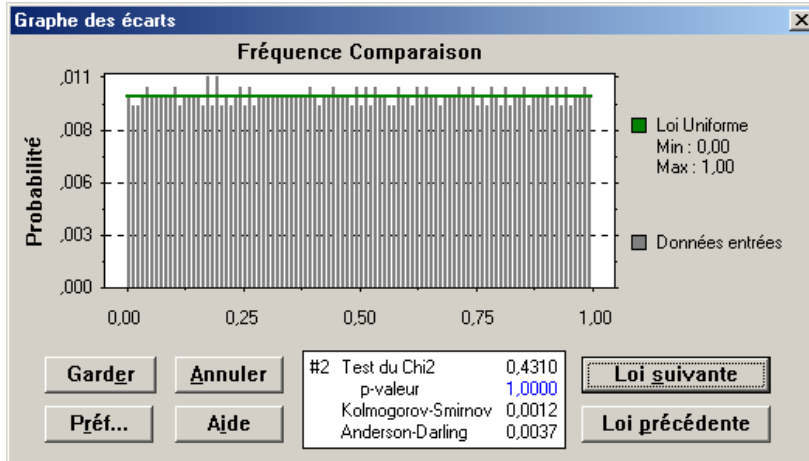
Exemple : Nous avons procédé à la même batterie de tests que ceux effectués sur la fonction *Rnd* de Visual Basic

✓ Simulation *Tore* avec 1 000 valeurs

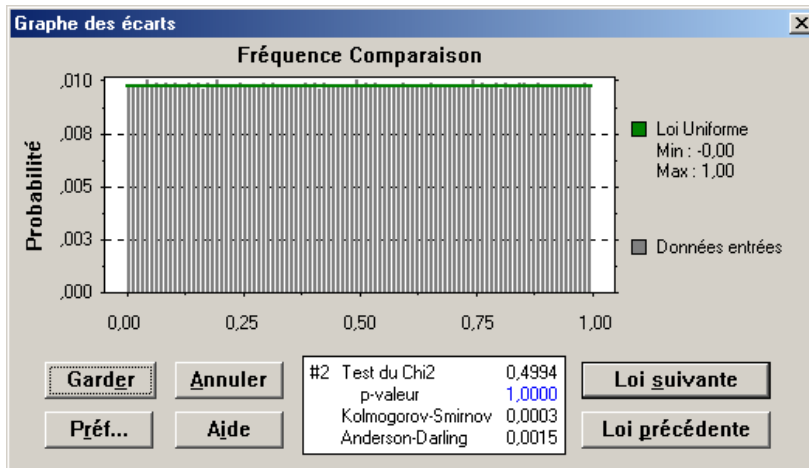


D'emblée, la répartition est bien meilleure que celle observée avec la fonction *Rnd*, elle paraît déjà très bien répartie malgré le nombre faible de valeurs.

✓ Simulation *Tore* avec 2 000 valeurs



✓ Simulation *Tore* avec 10 000 valeurs



Ici, la perfection de l'équirépartition sur $[0;1]$ est presque atteinte, en tout cas graphiquement. Même les tests ont des résultats excellents ; les P-valeurs du test du Chi-deux sont, dans les trois situations, égales à 1 !

3.2.3 Comparaison des deux types de générateurs

3.2.3.1 Exemple : valorisation d'une option d'achat européenne avec dividendes dans le cadre du modèle de Black et Scholes

L'objet de cet exemple est de mettre en pratique différents algorithmes de génération de nombres aléatoires de loi uniforme sur l'intervalle $[0;1]$, qu'ils soient pseudo ou quasi-

aléatoires, afin de valoriser « stochastiquement » une option d'achat européenne, et de comparer les résultats obtenus par simulations avec le résultat « analytique » donné par la formule de Black et Scholes. Rappelons tout d'abord brièvement le modèle de Black et Scholes en présence d'un versement continu de dividendes.

3.2.3.1.1 Présentation du modèle – Résolution analytique du problème posé

Ce modèle permet de valoriser de manière analytique une option d'achat européenne avec dividendes.

Hypothèse : Le rendement de l'actif sous jacent de l'option suit l'équation :

$$\frac{dS_t}{S_t} = (\mu - \delta) dt + \sigma dW_t ,$$

où : W_t est un brownien géométrique standard,

μ est l'espérance mathématique du rendement instantané de l'action ($\mu \in \mathbb{R}^+$),

σ l'écart type du rendement instantané de l'action ($\sigma \in \mathbb{R}^+$, σ constant),

δ le taux instantané de versement de dividendes.

Théorème : Formule de Black et Scholes avec versement continu de dividendes

Soit t l'instant étudié.

Soit S_t la valeur de l'actif sous jacent à l'instant t , en notant S la valeur de l'actif sous jacent à la date d'origine de l'option ($S = S_0$).

Soit K le prix d'exercice de l'option, et T sa durée de vie,

La valeur de l'option $C(S_t, t, K, T, \sigma, q)$ est donnée par :

$$C(S_t, t, K, T, \sigma, q) = S e^{-qt} \Phi(d_1) - K e^{-rT} \Phi(d_2)$$

où $\tau = T - t$,

$$d_1 = \frac{\ln(S/K) + (r - q + \frac{1}{2}\sigma^2) \times \tau}{\sigma \sqrt{\tau}},$$

$$d_2 = \frac{\ln(S/K) + (r - q - \frac{1}{2}\sigma^2) \times \tau}{\sigma \sqrt{\tau}},$$

$\Phi(d) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^d e^{-\frac{z^2}{2}} dz$, la fonction de répartition de la loi Normale centrée

réduite.

Il est ainsi possible de calculer de manière analytique le résultat de ce modèle, modulo l'application de la fonction $N(\cdot)$ qui est programmée par défaut dans *Excel*.

3.2.3.1.2 Résolution « stochastique » du problème posé

Dans cette approche, nous allons simuler un grand nombre N de trajectoires du cours de l'action sous la probabilité risque-neutre (à partir du processus de rendement log-normal, qui est l'une des hypothèses du modèle), et ce afin d'obtenir plusieurs valeurs possibles du cours du sous jacent à l'échéance de l'option. A chaque cours simulé, il sera possible de calculer la valeur à l'échéance de l'option $(S_T - K)_+$. Un estimateur sans biais de la véritable valeur à l'échéance de cette option sera la moyenne empirique sur l'ensemble des simulations des valeurs obtenues.

Enfin pour obtenir la valeur à la date des calculs de l'option, il suffira d'appliquer à cette moyenne des valeurs à l'échéance, un facteur d'actualisation $e^{-r\tau}$.

Formalisation : Notons :

U_i les nombres aléatoires de loi uniforme simulés,

$Norm$ la fonction de calcul des réalisations d'une loi Normale (0;1), à partir d'un nombre aléatoire de loi uniforme (fonction de type inversion de la fonction de répartition)

Alors la valeur simulée de l'option à la date des calculs t , est :

$$C(S, t, K, T, \Sigma) = \frac{e^{-r\tau}}{N} \sum_{i < N} (S_i e^{((r-d-\sigma^2/2)\tau + \sigma\sqrt{\tau}Norm(U_i))} - K)_+ .$$

En utilisant un très bon algorithme d'inversion de la loi Normale, l'algorithme de *De Moro* (unanimentement reconnu comme tel, et présenté par la suite), l'incertitude qui peut résider sur la qualité de l'évaluation de l'option vient de la qualité de la répartition des nombre aléatoires de loi Uniforme simulés sur l'intervalle [0;1]. Ainsi, comparer les différents types d'algorithmes étudiés ci-dessus avec la solution analytique du problème permet de « tester » leurs qualités respectives.

3.2.3.1.3 Application numérique

Nous cherchons à évaluer une option d'achat européenne dont les propriétés sont les suivantes :

- Taux d'intérêt sans risque : 8%,
- Volatilité du sous jacent : 25%,
- Taux de dividendes : 8%,
- Cours du sous jacent : 20 €,
- Prix d'exercice de l'option : 10 €,
- Temps restant avant l'expiration : 1 an.

La valeur analytique de cette option à la date actuelle est :

$$C(20;0;10;1;0,25) = 9,2239 \text{ €.}$$

Nous présentons ici trois types de résultats liés :

- ✓ aux simulations effectuées à partir de suites de *Van der Corput en base 2 (VdC)*,
- ✓ aux simulations effectuées sur des suites « *translation irrationnelle du Tore* » (*Tore*),
- ✓ aux simulations effectuées avec le *Rnd* de *Visual Basic (Rnd)*.

Chacun des simulateurs est utilisé avec l'algorithme d'inversion de la loi Normale de *De Moro*.

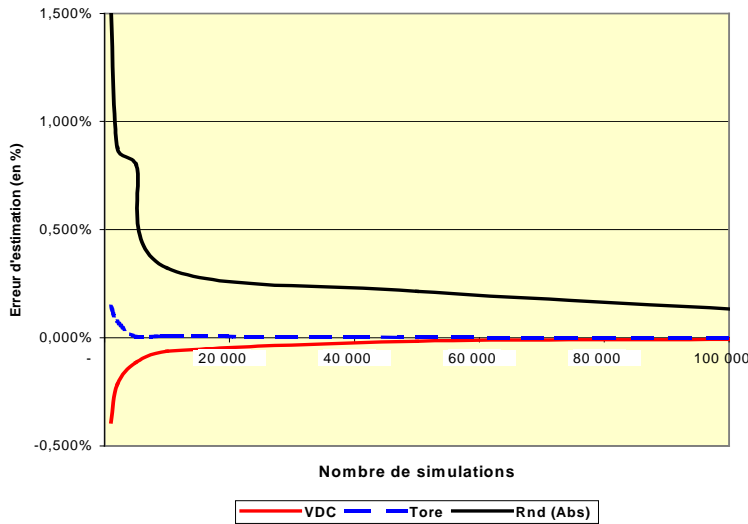
Le tableau ci dessous présente les écarts calculés par rapport à la valeur analytique, pour chaque type de générateurs aléatoires.

Nb : Lors de simulations quasi-aléatoires, c'est toujours la même valeur de l'option qui est générée, pour un même nombre de cours simulés, par propriété même de ces suites. Par contre, pour les simulations pseudo-aléatoires (Rnd ici), le prix de l'option obtenu n'étant pas toujours le même (notion d'initialisation de ce type de suite), le tableau suivant présente une moyenne des écarts relatifs (Moy), une moyenne des valeurs absolues des écarts relatifs (Moy Abs), le minimum et le maximum obtenus, le tout sur 10 simulations par valeur de N.

N	VdC	Tore	Rnd			
			Moy	Moy Abs	Min	Max
1 000	-0,389%	0,146%	0,036%	1,520%	-3,447%	2,760%
2 000	-0,215%	0,076%	-0,042%	0,879%	-1,393%	1,473%
5 000	-0,114%	0,006%	0,110%	0,802%	-1,713%	1,418%
10 000	-0,062%	0,008%	0,234%	0,323%	-0,327%	0,864%
50 000	-0,016%	0,002%	-0,091%	0,215%	-0,548%	0,365%
100 000	-0,008%	0,001%	0,028%	0,134%	-0,252%	0,342%

Graphiquement, les écarts obtenus en fonction du nombre de simulations pour chaque algorithme montrent la qualité de l'équirépartition des nombres de loi uniforme obtenus :

Comparaison de la qualité de la prévision en fonction du nombre de simulations



3.2.3.1.4 Conclusion

D'une manière générale, l'étude du tableau et du graphique précédents montre que les générateurs quasi-aléatoires (*VDC*, *Tore*) donnent des résultats qui convergent plus rapidement vers la solution analytique que le générateur pseudo-aléatoire (*Rnd*). Cette observation est d'ailleurs l'illustration pratique simple de la théorie de la discrétion, elle justifie visuellement le fait que la discrétion des suites quasi-aléatoires est plus faible par construction que celle des suites pseudo-aléatoires. L'approximation ainsi effectuée est plus rapidement correcte, et de plus, il n'y a pas besoin d'effectuer plusieurs simulations pour obtenir un résultat valable.

3.2.3.2 Remarques générales quant à la pertinence de l'utilisation de l'un ou l'autre des générateurs aléatoires de loi uniforme

Ce genre de considérations fait souvent l'objet de débats entre théoriciens et praticiens. En premier lieu, l'utilisation de suites totalement déterministes peut paraître plutôt « saugrenue » pour simuler des lois de probabilité, par essence non déterministes. Cependant, à la vue des résultats de la partie précédente, il semble que la question de l'utilisation de telles suites se pose réellement. En effet, si la convergence est bien plus rapide avec des simulations quasi-aléatoires, c'est parce que ces suites sont extrêmement mieux réparties sur $[0;1]^s$. Or une telle propriété ne peut être qu'un atout pour les domaines où la queue de distribution est très importante (comme la réassurance). Enfin il faut noter

que la rapidité de convergence est de nature à alléger le temps de calcul informatique parfois très conséquent.

4. CAS DES PROCESSUS ALÉATOIRES

Dans de nombreuses situations pratiques la modélisation ne repose pas uniquement sur des variables aléatoires discrètes, mais sur des processus stochastiques dont la dynamique est décrite par des équations différentielles stochastiques ; c'est notamment le cas de la plupart des modèles d'évolution de titres financiers.

L'objet de cette section est d'attirer l'attention du lecteur sur les particularités associées à ces modélisations, sans prétendre les traiter de manière exhaustive, ce qui fera l'objet d'un prochain article. Nous nous bornons donc à illustrer les difficultés propres aux cas des processus de diffusion sur la base d'un exemple.

Ainsi, dans le cas classique du modèle de Black et Scholes rappelé dans le présent article, le sous-jacent est supposé suivre un mouvement brownien géométrique décrit par l'EDS suivante :

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t.$$

La simulation de trajectoires du processus S nécessite, une fois le pas de discrétisation fixé, de définir un « équivalent discret » de l'équation ci-dessus ; la transposition au contexte des équations différentielles stochastiques des méthodes classiques de résolution numérique des équations différentielles ordinaires conduit en général à proposer une discrétisation basée sur le « schéma d'Euler », soit :

$$\frac{S_t - S_{t-h}}{S_{t-h}} = \mu * h + \sigma * \sqrt{h} * \varepsilon_{t-h},$$

avec (ε_t) un bruit blanc gaussien centré réduit et des instants t de la forme $n*h$, n entier.

On peut montrer que la vitesse de convergence d'un tel schéma de discrétisation vers le processus S est en \sqrt{h} , h étant le pas de discrétisation.

D'autres techniques spécifiques aux EDS peuvent être utilisées, comme le schéma de Milstein, afin d'améliorer la vitesse de convergence vers le processus continu. Ce schéma conduit à la discrétisation suivante :

$$\frac{S_t - S_{t-h}}{S_{t-h}} = \left(\mu - \frac{\sigma^2}{2}\right) * h + \sigma * \sqrt{h} * \varepsilon_{t-h} + \frac{\sigma^2 h}{2} * S_{t-h} * \varepsilon_{t-h}^2.$$

Avec un tel schéma on obtient une vitesse de convergence en h^2 .

La limitation du schéma de Milstein est qu'il n'en n'existe pas de version multivariée. Il s'agit d'une limitation importante car les modèles élaborés de modélisation de l'actif doivent intégrer les corrélations entre les différentes sources de risque.

Plus généralement, l'existence d'un biais de discrétisation liée à l'utilisation d'une approximation discrète d'un modèle initialement exprimé en temps continu doit être prise en considération ; le lecteur trouvera une illustration des méthodes de réduction de ce biais dans [18].

5. QUELQUES INDICATIONS SUR LA PRÉCISION DES RÉSULTATS.

Un bon algorithme de simulation devra donner une espérance empirique de X identique à celle obtenue par un calcul déterministe. Souvent, le choix du nombre de simulations K sera déterminant dans cette optique, car de manière générale, plus K est grand, plus l'espérance de X simulée est proche de la grandeur estimée, d'après la loi des grands nombres. Les théorèmes de grande déviation, au premier rang desquels on trouve le théorème central limite, fournissent des indications sur cet écart (voir par exemple [6] pour les résultats de base sur ce sujet).

Ainsi, l'erreur relative de la simulation peut être évaluée en fonction du nombre de simulations, ou inversement, fixer un niveau d'erreur maximal admissible (pour une probabilité fixée), permet de déterminer le nombre minimum de simulations à effectuer.

Cela permet notamment de vérifier dans toute étude faisant appel à des méthodes de simulation que cette technique est appropriée ; en effet, dans certains cas, on s'aperçoit qu'il faut un nombre de simulations presque infini pour n'avoir une erreur relative que très médiocre, comme l'illustre l'exemple ci-après...

Exemple : On choisit une loi définie par

$$X = \exp(\beta G),$$

avec G normale centrée réduite.

Le théorème central limite donne :

$$\frac{X_1 + \dots + X_n}{n} - E(X) \approx \frac{\sigma}{\sqrt{n}} G',$$

avec X_1, \dots, X_n n réalisations indépendantes de loi la loi de X , et G' gaussienne centrée réduite.

L'erreur relative est donnée par :

$$\frac{\frac{X_1 + \dots + X_n}{n} - E(X)}{E(X)} \approx \sqrt{\frac{\exp(\beta^2) - 1}{n}}.$$

Application numérique :

Avec $\beta = 1$, 1000 simulations permettent d'obtenir une erreur relative de 10 % environ. En revanche, avec $\beta = 5$, il faut 7×10^{10} simulations pour une erreur relative de 100 % !

On voit que si l'utilisation des méthodes de simulation est souple et pratique, et donne souvent de très bons résultats, il faut parfois prendre un recul nécessaire face à ces méthodes, afin de ne pas avoir des résultats biaisés.

Par ailleurs, il est toujours souhaitable de faire une validation a posteriori des simulations effectuées par le biais de tests statistiques du type test du chi-deux, ou éventuellement (avec une précision moindre) de tests graphiques (QQ plots par exemple) sur l'échantillon généré (cf. [6]).

Dans les cas « standards » il est souhaitable d'étalonner le modèle par simulation sur une sous-partie pour laquelle une solution explicite fermée est connue, ce qui permet de déterminer le nombre minimal de tirages nécessaire pour obtenir un niveau de précision donné (pour une probabilité fixée a priori).

6. CONCLUSION

Avant de conclure...

Nous tenons avant de conclure à attirer l'attention du lecteur sur un aspect très important que toute démarche s'appuyant sur des méthodes de simulation doit intégrer l'estimation des paramètres du modèle. Nous avons en effet fait ici l'hypothèse que les paramètres de base des phénomènes à simuler étaient bien déterminés, et cette étape de détermination et d'estimation des paramètres du modèle est en pratique délicate.

A titre d'illustration dans le domaine des processus de taux d'intérêt, le lecteur intéressé pourra se reporter à [18], qui propose des méthodes de réduction du biais de discrétisation dans l'estimation des paramètres d'un modèle de diffusion.

... pour conclure

Après avoir montré les limites des algorithmes de génération de nombres aléatoires « standards » intégrés aux logiciels utilisés pour les évaluations actuarielles, nous avons tenté dans cet exposé de présenter des techniques de simulation utilisables en pratique, aisées à mettre en oeuvre et améliorant sensiblement la performance et la précision des générateurs.

Ces techniques doivent faire partie aujourd'hui du « bagage technique » de l'actuaire pour mettre en application rigoureusement les modélisations de plus en plus complexes sur lesquelles reposent les évaluations actuarielles (DFA, valorisation de passifs sociaux en norme IFRS, ALM, etc.).

L'efficacité de ces méthodes est illustrée dans la seconde partie de cet article, consacré à deux exemples d'application.

7. BIBLIOGRAPHIE

OUVRAGES

- [1] KNUTH D.E., *The art of computer programming volume 2, Seminumerical Algorithms*. Book News, inc. (1997 – troisième édition).
- [2] BOWERS, GERBER, HICKMAN, JONES, NESBITT, *Actuarial Mathematics*. Society of Actuaries, (1997 – Seconde édition).
- [3] PARTRAT C., BESSON J.L., *Assurance Non Vie : Modélisation, Simulation*. Economica, (2005).
- [4] PETAUTON P., *Théorie et pratique de l'Assurance-Vie*. Dunod, (1996 – Seconde édition).
- [5] SAPORTA G., *Probabilité, Analyse de données et Statistiques*. Technip, (1990).
- [6] DROESBEKE J.J., FINE J., SAPORTA G., *Méthodes bayésiennes en statistiques*, Technip, (2002).

ARTICLES

- [7] MARSAGLIA, BRAY, *A convenient method for generating Normal variables*. SIAM Review, n°4 (1964).
- [8] GE Z., *A numerical study of one-factor interest rate models*. A thesis submitted at the University of Toronto, n°4 (1998).
- [9] ROBERTSON J.P., *The computation of aggregate loss distributions*. Proceedings of the Casualty Actuarial Society, Vol. LXXIX, Pages 57-133 (1992).
- [10] HECKMAN P.E., MEYERS G.G., *The calculation of aggregate loss distributions from claim severity and claim count distributions*. Proceedings of the Casualty Actuarial Society, Vol. LXX, Pages 22-61 (1983).
- [11] GREENBERGER M., *Notes a new pseudo-random number generator*. Journal of the ACM, 8, (2), April, p. 163-167, (1961).

- [12] LUBES-NEIL H., MASSON J.L., CAPPELAERE B., *Quelques techniques de simulations de variables aléatoires applicables à l'hydrologie*. UMR HydroSciences, Montpellier.
- [13] SCOLLNIK D.P., *An introduction to Markov Chain Monte Carlo methods and their actuarial applications*. Proceedings of the Casualty Actuarial Society, Vol. LXXXIII, Pages 114-165 (1996).
- [14] KAUFMANN R., GADMER A. et KLETT R., *Introduction to Dynamic Financial Analysis*. Astin Bulletin, Vol. 31, Pages 213-249, n°1 (2001).
- [15] AHLGRIM K.C., D'ARCY S.P. et GORVETT R.W., *Parametrizing Interest Rate Models*. Casualty Actuarial Society Forum, Pages 1-50 (1999).
- [16] ROGERS L.C.G., *Which modelo for term-structure of interest rates should one use?* Mathematical Finance, IMA Vol. 65, Springer, p. 93-116 (1995).
- [17] DE WINNE R., *Processus de diffusion de taux d'intérêt et correction du biais de discrétisation*. 5th AFIR International Colloquium, p. 749 – 782.

THESES - MEMOIRES

- [18] PATARD P.A., *Valorisation de SWAPS structures sur actions et indices*. Mémoire confidentiel présenté devant le jury de l'Institut de Science Financière et d'Assurances en janvier 2002.
- [19] MAGNIN F., *L'engagement de l'employeur dans le cadre d'un régime de retraite à prestations définies*. Mémoire confidentiel présenté devant le jury de l'Institut de Science Financière et d'Assurances en 1998.
- [20] LEMIEUX C., *L'utilisation des règles de réseau en simulation comme technique de réduction de la variance*. Thèse présente à la Faculté des études supérieures (Université de Montréal) en vue de l'obtention du grade de Philosophæ Doctor en informatique.
- [21] JAL P., *Cotation XS en réassurance short - tail*. Mémoire confidentiel présenté devant le jury de l'Institut de Science Financière et d'Assurances en 2001.
- [22] JACQUEMIN J., *Les méthodes de simulation en assurance*. Mémoire confidentiel présenté devant le jury de l'Institut de Science Financière et d'Assurances en 2002.

NETOGRAPHIE

- [23] www.arlj.org/docs/algo/random.php, *Produire des nombres "au hasard"*.
- [24] <http://rfv.insa-lyon.fr/~jolion/STAT/node1.html>, *Probabilités et statistiques*.

ANNEXE : CONSTANTES DE L'ALGORITHME DE DE MORO

Voici ces valeurs telles que déclarées dans l'algorithme de calcul :

$$a(0) = 2.50662823884$$

$$a(1) = -18.6150062529$$

$$a(2) = 41.39119773534$$

$$a(3) = -25.44106049637$$

$$b(0) = 1$$

$$b(1) = -8.4735109309$$

$$b(2) = 23.08336743743$$

$$b(3) = -21.06224101826$$

$$b(4) = 3.13082909833$$

$$c(0) = 7.71088707054879$$

$$c(1) = 2.77720135336852$$

$$c(2) = 0.3614964129261$$

$$c(3) = 3.73418233434554E-02$$

$$c(4) = 2.8297143036967E-03$$

$$c(5) = 1.625716917922E-04$$

$$c(6) = 8.017330474 * 10 ^ (-6)$$

$$c(7) = 0.3840919865 * 10 ^ (-6)$$

$$c(8) = 0.012970717 * 10 ^ (-6)$$

$$k(0) = 0.417988642492643$$

$$k(1) = 4.24546868813766$$