Méthodes de Monte Carlo

appliquées au pricing d'options

et à la gestion des risques multiples

Cours ENSAI de $3^{\text{ème}}$ année

Nicolas BAUD

Notes de cours écrites avec la collaboration de Thierry Roncalli

Groupe de Recherche Opérationnelle
Crédit Lyonnais
Bercy-Expo – Immeuble Bercy Sud – 4ème étage
90, Quai de Bercy — 75613 Paris Cedex 12
France

nicolas.baud@creditlyonnais.fr

Table des matières

1 Simulation d'une loi uniforme **1**

1.1 Générateurs congruentiels linéaires 2

1.2 Tests d'uniformité 3

1.2.1 Structure latticielle des générateurs congruentiels 4

1.2.2 Tests statistiques d'uniformité 6

1.3 Autres types de générateurs 8

1.3.1 Générateurs à déplacement de registre 8

1.3.2 Combinaison de générateurs : le cas du générateur KISS 9

2 Simulation d'une loi non uniforme **11**

2.1 Tests d'adéquation à une loi donnée 11

2.1.1 Test du Khi-deux 11

2.1.2 Test de Kolmogorov-Smirnov 12

2.1.3 Test de Cramer-von Mises 12

2.2 Méthode des transformations 13

2.2.1 Inversion de la fonction de répartition 13

2.2.2 Utilisation de Relations fonctionnelles 15

2.3 Le cas des distributions discrètes 17

2.3.1 Méthode d'inversion, le cas général 18

2.3.2 Inversion à partir de la distribution d'une loi continue 18

2.3.3 La méthode de la table guide 19

2.3.4 Le cas particulier de la loi de Poisson 20

2.4 La méthode des mélanges discrets 21

2.5 Simulation par acceptation-rejet 22

2.5.1 Le cas général 22

2.5.2 La méthode du "squeeze" 25

2.5.3 Le cas des densité log-concaves : La méthode des cordes 26

2.5.4 La méthode du rapport d'uniformes. 27

3 Méthodes de Monte Carlo appliquées au pricing d'option : calcul d'intégrales, accélération de la convergence **29**
3.1 Contrôle de la convergence 30
3.2 Variables de contrôle 31
3.3 Échantillonnage d'importance 35
3.3.1 Méthode générale 35
3.3.2 Le cas des processus de diffusion 39
3.4 Stratification 42
3.5 Variables antithétiques 44

4 Utilisation des copules pour la simulation d'un vecteur aléatoire **45**
4.1 Simulation de variables aléatoires uniformes indépendantes 45
4.2 La méthode des distributions 46
4.3 La méthode des distributions conditionnelles 51
4.4 Les méthodes dites analytiques 56
4.5 La méthode des quantiles empiriques 57
4.6 Repères historiques 58

5 Méthodes de Monte Carlo appliquées à le gestion des risques **59**
5.1 Le problème de l'agrégation de (et dans) la valeur en risque 59
5.2 Différentes méthodes de construction la VaR de marché 63
5.2.1 VaR historique 64
5.2.2 VaR gaussienne 64
5.2.3 VaR paramétrique 65
5.2.4 VaR semi-historique 68
5.3 Calcul de la charge en capital pour le risque opérationnel 70

1

Simulation d'une loi uniforme

Les méthodes de simulation reposent toutes sur la capacité à produire une suite de variables aléatoires indépendantes suivant une distribution f donnée. La production d'une telle suite dépend elle-même de la capacité que l'on a à produire une séquence de variables aléatoires uniformes et indépendantes.

En pratique, la production par des moyens informatiques d'une suite de variables uniformes iid est impossible : toute suite de nombres résulte de la mise en œuvre d'un algorithme, et par conséquent est déterministe. L'objectif que l'on s'assigne n'est donc pas la production de véritables variables aléatoires, mais la production de séquences déterministes imitant le mieux possible le comportement d'une suite de variables aléatoires uniformes iid : on parle de nombres pseudo-aléatoires.

L'utilisation de tout algorithme destiné à produire des suites de nombres aléatoires uniformément répartis sur $[0, 1]$ doit donc être validée par la mise en œuvre de tests statistiques dont l'hypothèse nulle H_0 est :

$$H_0 : U_1,U_2,...,U_N \text{ sont iid de loi uniforme sur } [0, 1].$$

On peut pour cela considérer des tests classiques d'adéquation à une loi donnée (test du Khi-deux, test de Kolmogorov-Smirnov, test de Cramer-von Mises) ou tout autre type de tests reposant sur H_0. On peut par exemple utiliser les tests de la batterie Diehard (disponible sur http ://stat.fsu.edu/˜geo).

Dans la pratique, un générateur de nombres pseudo-aléatoire peut sembler correct car validé par un ensemble de tests, mais il est toujours possible de construire un nouveau test invalidant l'hypothèse d'uniformité. Il convient donc d'être pragmatique lors du choix d'un générateur de nombres pseudo-aléatoire : ce choix doit tenir compte des caractéristiques de l'application envisagée.

Dans la suite de ce chapitre, nous nous intéressons dans un premier temps aux générateurs congruentiels linéaires. Ce type de générateur est fréquemment rencontré car :
- sa mise en œuvre est très simple ;
- il présente des propriétés statistiques satisfaisantes pour la plupart des applications classiques.

Dans un deuxième temps, nous exposons quelques exemples de tests statistiques pouvant être menés pour valider l'utilisation d'un générateur de nombres pseudo-aléatoires.

La troisième partie de ce chapitre présente succinctement une autre méthode permettant de simuler une suite de nombre pseudo-aléatoires.

1.1 Générateurs congruentiels linéaires

Dans cette première section, on se propose d'étudier les principales caractéristiques des générateurs congruentiels linéaires. Ce type de générateur produit des suites de nombres pseudo-aléatoires sur $[0, 1]$ à partir de l'algorithme suivant :

Algorithme 1.1 – *Générateur congruentiel linéaire* –

Étant donnés deux paramètres entiers A et M,

1. *Choisir une racine x_0*
2. *à l'étape $n > 0$, construire $x_n = Ax_{n-1}\,[M]$ et retourner $u_n = \frac{x_n}{M}$*

Il apparaît ici clairement que ce type de générateur est périodique, de période P strictement inférieure à M. Si l'on souhaite s'approcher le plus possible d'une distribution continue uniforme sur $[0, 1]$, il convient donc de choisir A et M de sorte que la période P soit la plus grande possible.

Choix des paramètres A et M

Dans le cas où M est un nombre premier, le choix de A repose sur le résultat suivant :

Théorème 1.1 *Si M est un nombre premier, alors la suite $x_n, n \geq 0$, obtenue à partir de l'algorithme 1.1 est de période $M - 1$ si et seulement si A est une racine primitive de M :*

$$A^{M-1} \equiv 1\,[M] \;\; et \;\; A^{k-1} \neq 1\,[M]\,, \forall k < M$$

Le générateur de nombres pseudo-aléatoires $rndu(.,.)$ utilisé par le logiciel Gauss exploite ce résultat. Les paramètres M et A prennent les valeurs suivantes :

$$\begin{cases} M_G = 2^{31} - 1 \\ A_G = 397204094 \end{cases}$$

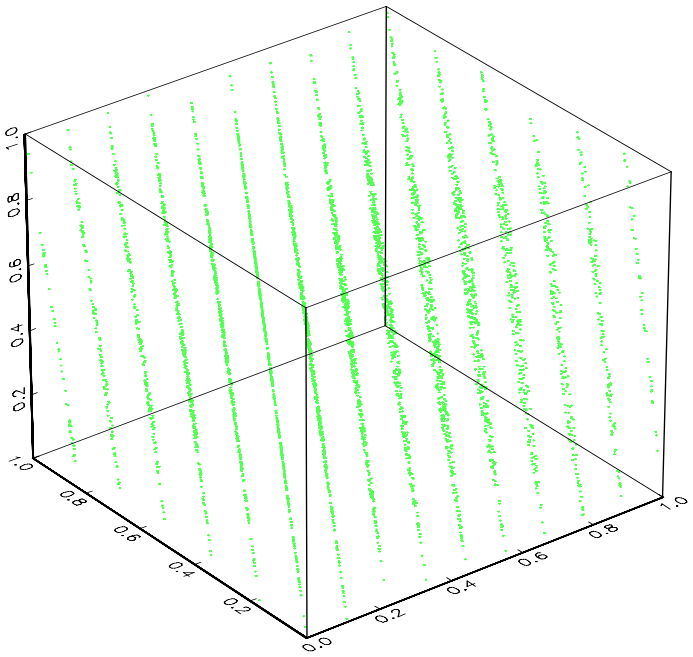
Dans la mesure où un générateur de nombres pseudo-aléatoires est très fréquemment appelé au cours d'une expérience de Monte Carlo, un autre critère intervenant dans le choix de M est le temps de calcul qu'il induit pour chaque élément de la suite (u_n) produite par l'algorithme 1.1. La représentation interne des nombres sous la forme de séquences binaires conduit à s'intéresser aux paramètres M s'écrivant sous la forme $M = 2^\beta$:

- la division modulo M consiste alors à mettre à 0 l'ensemble des bits de rang supérieur à β ;
- la division par M est réalisée en décalant le bit représentatif de la virgule de β bits vers la gauche.

Dans le cas où M est de la forme 2^β, le théorème suivant caractérise les coefficients multiplicatifs A permettant d'obtenir une suite $x_n, n \geq 0$, de période maximale :

Théorème 1.2 *Si $M = 2^\beta, \beta \geq 3$, la période maximale $P = 2^{\beta-2}$ est réalisée si et seulement si $A \equiv \pm 3[8]$ et $x_0 \equiv 1[8]$.*

Les deux théorèmes que nous venons d'énoncer dans le cas où M est premier ou de la forme 2^β permettent de déterminer les sous-ensembles d'entiers auxquels le paramètre A doit appartenir si l'on souhaite obtenir un générateur de période la plus grande possible. Cette dernière propriété assure la production d'une suite de nombres pseudo-aléatoires dont la distribution est proche d'une distribution uniforme sur $[0, 1]$. Néanmoins, si la dimension du problème augmente et que l'on souhaite approcher une distribution uniforme sur $[0, 1]^k, k \geq 2$, rien n'assure que l'utilisation d'un coefficient multiplicatif A déterminé sur la seule base du théorème 1.1 dans le cas où M est



Graphique 1.1. Structure latticielle du générateur RANDU

premier, ou du théorème 1.2 dans le cas où M est une puissance de 2, permette de produire une suite présentant des propriétés proches de celles de lois uniformes iid sur $[0,1]^k$.

A titre d'exemple, on peut citer le cas du générateur RANDU. Ce générateur utilise les coefficients A et M suivants :

$$\begin{cases} M = 2^{32} \\ A = 65539 = 2^{16} + 3. \end{cases}$$

M est ici de la forme 2^β et $A \equiv 3[8]$ ce qui assure, compte tenu du théorème 1.2 page ci-contre, que la suite de nombres $u_n \in [0,1], n \geq 0$, obtenue à partir de RANDU est de période maximale. Cependant, on peut facilement montrer que (u_n) vérifie :

$$u_{n+1} - 6u_n + 9u_{n-1} \equiv 0[1], \forall n.$$

Or, puisque $\forall n, u_n \in]0,1[$, on a :

$$-6 < u_{n+1} - 6u_n + 9u_{n-1} < 10,$$

si bien que les triplets (u_{n-1}, u_n, u_{n+1}) produit par RANDU sont situés sur au plus 15 plans (cf graphique 1.1) ! Une telle régularité va bien entendu à l'encontre des propriétés attendues pour un générateur de nombres pseudo-aléatoires.

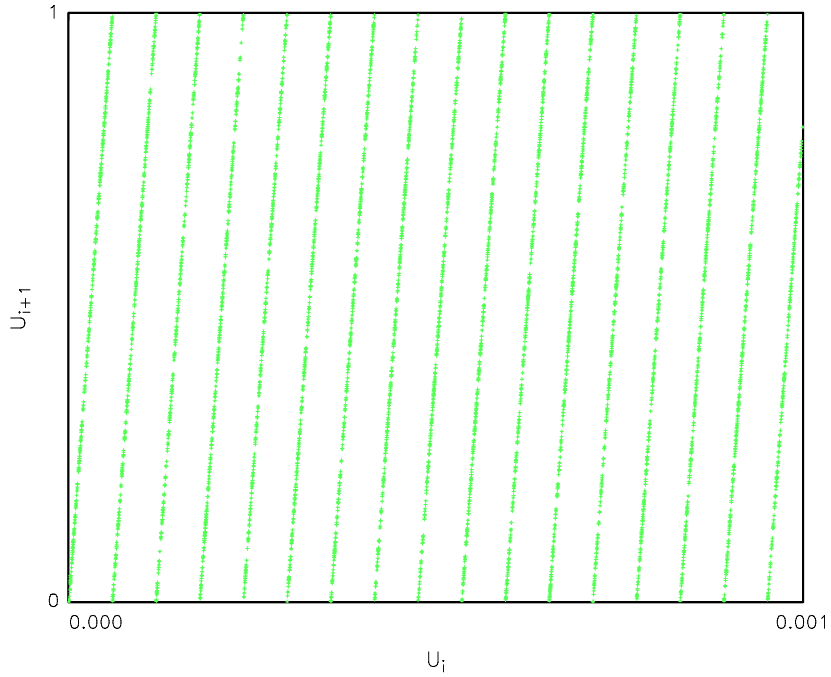
1.2 Tests d'uniformité

Il existe au moins deux types de tests d'uniformité différents pour les générateurs congruentiels linéaires. Le premier type de tests est spécifique des générateurs congruentiels, fondé sur la structure latticielle des nombres qu'ils produisent. Dans le cas du générateur RANDU, nous venons de voir

qu'en dimension $k = 3$, les triplets (u_n, u_{n+1}, u_{n+2}) se localisent sur un nombre fini d'hyperplans. Une telle propriété se généralise à l'ensemble des générateurs congruentiels linéaires, quelle que soit la dimension k considérée.

Le second type de tests d'uniformité est plus général puisqu'il est adapté à n'importe quel générateur de nombres pseudo-aléatoires : il s'intéresse aux propriétés statistiques que doivent vérifier des nombres aléatoires s'ils sont indépendants et distribués suivant une loi uniforme.

1.2.1 Structure latticielle des générateurs congruentiels



Graphique 1.2. Représentation de (U_i, U_{i+1}) dans le cas où $A = 16807$ et $M = 2^{31} - 1$

Dans le cas où M est premier[1], tous les k-uplets $(x_i, x_{i+1}, \ldots, x_{i+k-1})$ construits à partir de l'algorithme 1.1 page 2 se situent sur des familles d'hyperplans de la forme (Marsaglia, 1968)

$$\sum_{s=}^{k-1} q_s x_{i+s} \equiv 0 \, (mod \, (M)) \, .$$

Un tel résultat constitue un défaut pour les générateurs congruentiels : il met en évidence les propriétés de régularité de la répartition dans l'espace des nombres pseudo-aléatoires construits à partir de ce type de générateur. La régularité induite par un générateur congruentiel dépend bien entendu du choix des coefficients A et M.

Les deux paragraphes suivants présentent deux tests d'uniformité reposant sur la structure latticielle des générateurs congruentiels linéaires. A titre d'exemple, ces deux tests ont été appliqués à $rndu\,(.,.)$.

[1] Il existe des résultats équivalents dans le cas où M est de la forme 2^β. Le lecteur pourra se reporter à [8, Fishman] pour plus de détails.

Test spectral

Pour une famille d'hyperplans parallèles induite par un couple de paramètres (A, M) et caractérisée par un vecteur q, on considère la distance (euclidienne) entre deux hyperplans consécutifs, que l'on note $d_k(q, A, M)$. On définit alors

$$d_k(A, M) = \max_q d_k(q, A, M)$$

comme étant la distance maximale entre deux hyperplans parallèles adjacents parmi l'ensemble des familles d'hyperplans possibles.

Plus cette distance est petite, meilleur est le choix des paramètres A et M : le générateur couvre mieux l'hypercube $[0, 1]^k$. La minimisation de cette distance peut donc constituer un premier critère permettant de déterminer les paramètres A et M. Notons que $d_k(A, M)$ ne peut pas être rendu aussi petit qu'on le désire. Cette propriété fait l'objet du lemme suivant :

Lemme 1.1 *Soit M un nombre premier. Il existe γ_k tel que $d_k(A, M) M^{\frac{1}{k}} \geq \gamma_k$.*

On peut donc construire un premier critère normalisé S_1, défini par :

$$S_1 = \frac{\gamma_k}{d_k(A, M) M^{\frac{1}{k}}}.$$

S_1 doit être le plus proche de 1 possible.

Nombre d'hyperplans minimal

On deuxième critère caractérisant la distribution latticielle liée au couple (A, M) est le nombre d'hyperplans parallèles $N_k(q, A, M)$ contenant l'ensemble des k-uplets possibles : plus ce nombre est petit, plus la taille des régions de $[0, 1]^k$ qui sont vides est importante. Considérons alors la plus mauvaise performance $N_k(A, M)$ définie par

$$N_k(A, M) = \min_q N_k(q, A, M).$$

De même que dans la section précédente, on dispose d'un lemme montrant l'existence d'une borne supérieure pour $N_k(A, M)$, indépendante de A :

Lemme 1.2 *Soit M un nombre premier. $N_k(A, M) \leq (k!M)^{\frac{1}{k}}$.*

On peut donc définir un second critère normalisé S_2 par :

$$S_2 = \frac{N_k(A, M)}{(k!M)^{\frac{1}{k}}}$$

De même que pour S_1, le choix de A et M sera d'autant plus judicieux que la valeur de S_2 sera proche de 1.

Application : le cas de $rndu(.,.)$

Le tableau suivant présente les valeurs de S_1 et S_2 pour le générateur utilisé par Gauss en fonction de la dimension considérée :

Dimension	2	3	4	5	6	7	8
S_1	0.556	0.575	0.667	0.768	0.595	0.581	0.718
S_2	0.597	0.504	0.624	0.660	–	–	–

Ces valeurs peuvent être comparées, par exemple, à celles que l'on obtient dans le cas où $A = 742938285$ (cf. [8, Fishman]) :

Dimension	2	3	4	5	6	7	8
S_1	0.867	0.861	0.863	0.832	0.834	0.624	0.706
S_2	0.836	0.661	0.6662	0.602	–	–	–

Au vu de ces deux tableaux, il apparaît que le coefficient A_G n'est pas optimal pour les critères S_1 et S_2. Notons néanmoins que l'écart relatif des valeurs de S_1 et S_2 diminue avec la dimension, le critère S_2 devenant même favorable à A_G lorsque la dimension est égale à 5.

1.2.2 Tests statistiques d'uniformité

La seconde famille de tests à considérer est plus générale : elle peut s'appliquer à tout type de générateur de nombres pseudo-aléatoires. Il s'agit de tests exploitant les propriétés statistiques vérifiées par toute suite de variables aléatoires indépendantes, uniformément réparties sur $[0, 1]$. Chaque test consiste à vérifier que l'hypothèse nulle H_0 suivante est vérifiée :

$$H_0 : U_1, U_2, ..., U_N \text{ sont iid de loi uniforme sur } [0, 1].$$

Notons qu'il n'existe pas de test universel : les tests à retenir dépendent de l'application considérée. A titre d'exemple, nous présentons dans ce qui suit quatre tests issus de la batterie de tests proposée par G. Marsaglia : Diehard. Ces tests ont été appliqués à $rndu\,(.,.)$.

Test des Permutations : *the overlapping 5-permutation test*

Ce test considère l'ensemble des suites de 5 entiers consécutifs, chacune de ces suites constituant un "mot" de 5 lettres auquel on associe l'état dans lequel il se trouve parmi les $120 = 5!$ états possibles. Sur un échantillon donné, le nombre de "mots" observés dans chacun des états permet de construire un test du χ_2 asymptotique. Le tableau ci-dessous donne les résultats de ce test sur deux échantillons différents d'un million d'entiers obtenus à partir de $rndu(.,.)$:

$\chi_2\,(99)$	p-value
108.64	0.76
105.76	0.70

Au vu de ces deux résultats, $rndu(.,.)$ passe le test des permutations avec succès.

Tests de rang sur matrices binaires : *the binary rank tests*

L'ensemble des tests de rang repose sur la construction de matrices composées de 0 et de 1 à partir des valeurs observées sur les bits définissant chacun des entiers produit par le générateur de nombres pseudo-aléatoires étudié.

Matrices de taille 31

Les lignes des matrices sont construites à partir des 31 bits caractérisant chacun des entiers générés par $rndu\,(.,.)$. La construction d'une matrice nécessite donc au total 31 entiers. Le rang de chacune des matrices ainsi construite est compris entre 0 et 31. La comparaison de la distribution empirique à la distribution théorique est effectuée au travers d'un test du χ_2.

Le tableau ci-après donne les résultats observés sur un échantillon de 40000 matrices. Notons que l'observation de matrices de rang strictement inférieur à 28 étant très rare, ces matrices ont été rassemblées avec les matrices de rang 28.

Rang	observé	Espéré	$\frac{(o-e)^2}{e}$	Somme
28	226	211.4	0.005753	1.006
29	5115	5134	0.070391	1.076
30	23151	23103	0.099532	1.176
31	11508	11551.5	0.163993	1.340

La p-value associée à 1.340 pour un χ_2 à 3 degrés de liberté est 0.40. Le test de rang sur matrices de taille 31 est donc passé avec succès.

Matrices de taille 6×8

On construit une matrice de 6 lignes et 8 colonnes à partir de 6 entiers. Chacune des lignes correspond à 8 bits consécutifs choisis parmi les 32 bits définissant chacun des entiers utilisés. Pour chacune des 25 séquences de 8 bits possibles, on construit un test de rang semblable au test utilisé pour les matrices de taille 31. Le tableau ci-après présente les 25 p-values associées à l'ensemble de ces tests (réalisés sur 100000 matrices).

0.436709	0.738911	0.776722	0.302978	0.754293
0.103562	0.178121	0.802830	0.707586	0.491288
0.952403	0.112779	0.101678	0.405846	0.012914
0.493125	0.837334	0.303682	0.191681	0.902893
0.154640	0.500737	0.818182	0.270197	1.000000[2]

Les 25 tests étant indépendants, la distribution de ces p-values devrait être uniforme sur $[0, 1]$. La construction d'un test de Kolmogorov-Smirnov sur ces 25 p-values conduit à une p-value de 0.874963. *rndu* (., .) passe donc le test de rang sur matrices de taille 6*8 avec succès.

Test des séries : *the runs test*

Ce test s'intéresse au nombre de séquences croissantes et de séquences décroissantes produites par le générateur de nombres pseudo-aléatoires. Sur un échantillon donné, la distribution des séquences croissantes est comparée à la distribution théorique par l'intermédiaire d'un test du χ_2. Ce test est réalisé de la même façon pour les séquences décroissantes.

Le tableau ci-après donne les p-values associées aux tests de Kolmogorov-Smirnov réalisés sur 10 p-values obtenues à partir des tests du χ_2. Les échantillons utilisés sont de taille 10000.

	runs up	runs down
Test 1	0.567	0.020
Test 2	0.320	0.492

Le générateur *rndu* (., .) passe le test des séries avec succès.

Test de la sphère : *the 3DSPHERES Test*

Il s'agit d'un test en dimension 3. Ce test repose sur le protocole suivant :
- choisir aléatoirement 4000 points à l'intérieur d'un cube de côté 1000 ;
- associer à chacun de ces points pris comme centre d'une sphère le volume de la plus petite sphère permettant d'atteindre le plus proche voisin.
- retenir la plus petite des sphères parmi les 4000 sphères possibles.

Si les 4000 points sont tirés suivant une loi uniforme, de façon indépendante, le volume de cette dernière sphère est distribué suivant une loi exponentielle de moyenne $120\frac{\pi}{3}$. Le rayon au cube r^3 est donc distribué suivant une loi exponentielle de moyenne 30.

Le test *3DSPHERES* de Diehard effectue 20 expériences, ce qui permet d'obtenir 20 valeurs pour le plus petit rayon r^3. Un test de Kolmogorov - Smirnov est alors effectué sur les 20 p-values $1 - \exp\left(\frac{r^3}{30}\right)$, celles-ci devant être réparties uniformément sur $[0, 1]$.

Ce test, appliqué à un échantillon construit à partir de $rndu\,(.,.)$ conduit aux résultats suivants :

ÉCHANT.	R^3	P-VALUE	ÉCHANT.	R^3	P-VALUE
1	79.62	0.930	6	12.67	0.344
2	10.83	0.303	7	9.73	0.277
3	22.09	0.521	8	15.34	0.400
4	20.43	0.494	9	4.71	0.145
5	23.81	0.548	10	146.40	0.992

ÉCHANT.	R^3	P-VALUE	ÉCHANT.	R^3	P-VALUE
11	29.83	0.630	16	47.38	0.794
12	53.57	0.832	17	3.72	0.117
13	25.23	0.569	18	32.40	0.660
14	21.07	0.505	19	60.77	0.868
15	1.99	0.0640	20	6.03	0.182

La p-value associée au test de Kolmogorov - Smirnov est 0.015, ce qui ne permet pas de rejeter l'hypothèse d'uniformité.

1.3 Autres types de générateurs

Dans cette dernière section, nous présentons rapidement d'autres types de générateurs de nombres pseudo-aléatoires. En particulier, nous présentons la façon dont le générateur KISS est mis en œuvre : il s'agit d'une combinaison de trois générateurs "élémentaires" : un générateur congruentiel linéaire d'une part, et deux générateurs à déplacement de registre d'autre part.

1.3.1 Générateurs à déplacement de registre

Les générateurs à déplacement de registre reposent sur l'indépendance théorique des k composantes binaires d'un variable aléatoire X de loi uniforme sur $\{0, 1, ..., 2^k - 1\}$. L'algorithme général permettant de mettre en œuvre ce type de générateur est le suivant :

Algorithme 1.2 – *générateur à déplacement de registre* –

Etant donnée une matrice T de dimension $k \times k$ composée de 0 et de 1, et un module $M = 2^k - 1$,

1. *Choisir une racine x_0*
2. *à l'étape $n > 0$, construire $x_n = Tx_{n-1}[2]$, x_n étant ici représenté sous la forme de ces k composantes binaires $e_{n,i}$:*
$$x_n = \sum_{i=0}^{k-1} e_{n,i} 2^i,$$
3. *retourner $u_n = \frac{x_n}{M}$.*

L'intérêt de ce type de générateur est qu'il utilise directement la représentation interne des entiers sous forme de séquences de bits : les différentes opérations présentes dans l'algorithme précédent ne requièrent que très peu de temps de calcul. Pour plus de détails sur ce type de générateur, le lecteur pourra se reporter à [8, Fishman].

1.3.2 Combinaison de générateurs : le cas du générateur KISS

Une troisième technique permettant de construire des générateurs de nombres pseudo-aléatoires consiste à combiner des générateurs "simples", relativement peu performants, afin d'obtenir un générateur de meilleure qualité. Nous illustrons ici cette méthode au travers de l'exemple du générateur KISS.

La suite de nombres pseudo-aléatoires $X_1, ..., X_N$ produite par le générateur KISS est obtenue à partir de trois suites d'entiers. La première suite (I_n) correspond à un générateur congruentiel :

$$I_{n+1} = 69069 I_n + 23606797[2^{32}].$$

Les deux autres suites correspondent à des générateurs à déplacement de registre :

$$\begin{aligned} J_{n+1} &= \left(I + L^{15}\right)\left(I + R^{17}\right) J_n[2^{32}] \\ K_{n+1} &= \left(I + L^{13}\right)\left(I + R^{18}\right) K_n[2^{31}] \end{aligned}$$

où I représente l'opérateur identité, et R et L les opérateurs de décalage à droite et à gauche :

$$\begin{aligned} R(e_1, ..., e_k)^\top &= (0, e_1, ..., e_{k-1})^\top \\ L(e_1, ..., e_k)^\top &= (e_2, ..., e_k, 0)^\top \end{aligned}$$

X_{n+1} est alors obtenu en combinant les trois nombres I_{n+1}, J_{n+1} et K_{n+1} :

$$X_{n+1} = I_{n+1} + J_{n+1} + K_{n+1}[2^{32}].$$

La période du générateur ainsi obtenu est de l'ordre de 2^{95}, ce qui est nettement supérieur à la période de chacun des générateurs élémentaires intervenant dans la construction de la suite $X_1, .., X_N$ (ces périodes n'excèdent pas 2^{32}). Notons de plus que le générateur KISS passe avec succès l'ensemble des tests de la batterie Diehard.

2

Simulation d'une loi non uniforme

Ce chapitre présente les méthodes les plus usuelles utilisées pour simuler des variables distribuées suivant une loi non uniforme :

- La première de ces méthodes est la méthode des transformations. Elle repose sur l'utilisation d'une relation fonctionnelle entre la variable que l'on souhaite simuler et une ou plusieurs autres variables aléatoires "élémentaires", faciles à simuler.
- L'autre classe de méthodes abordée dans ce chapitre correspond aux méthodes d'acceptation-rejet. Ces méthodes consistent à simuler des variables aléatoires suivant une loi instrumentale "simple" et à ne retenir que les réalisations remplissant un certain critère : la règle d'acceptation utilisée permet de modifier la distribution des variables retenues de telle sorte qu'elles soient distribuées suivant la loi objectif que l'on souhaite simuler.

Au préalable, nous présentons trois tests classiques permettant de vérifier l'adéquation d'une distribution empirique à une loi données.

2.1 Tests d'adéquation à une loi donnée

Étant donnée une fonction de répartition F cible, on souhaite savoir si l'échantillon x_1,...,x_N produit par un générateur de nombres pseudo-aléatoires peut être considéré comme issu de cette loi. Si l'on note $\hat{F}$ la distribution associée au générateur, le test à mener considère $H_0 : \hat{F} = F$ contre $H_1 : \hat{F} \neq F$.

2.1.1 Test du Khi-deux

Les observations x_1,...,x_N sont réparties en K classes disjointes $[a_{i-1}, a_i]$. L'effectif empirique de chacune des classes est noté n_j. Chacun de ces effectifs est comparé à l'effectif théorique

$$\begin{aligned} Np_j &= Npr\left(X \in [a_{i-1}, a_i]\right) \\ &= N\left[F(a_i) - F(a_{i-1})\right]. \end{aligned}$$

Sous H_0,

$$D^2 = \sum_{i=1}^{K} \frac{(n_j - Np_j)^2}{Np_j} \underset{loi}{\rightarrow} \chi^2_{K-1} \text{ quand } N \rightarrow \infty$$

On rejettera H_0 lorsque la distance entre les fréquences observées et théoriques sera grande, ce qui conduit à la région critique de niveau α : $\left\{D^2 > c\right\}$ où $\alpha = pr\left(D^2 > c\right)$;

2.1.2 Test de Kolmogorov-Smirnov

Ce test est fondé sur l'utilisation de la fonction de répartition empirique F_N de l'échantillon x_1,...,x_N :

$$F_N(x) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\{x_i \leq x\}},$$

et sur l'utilisation de la distance $d_K(F_N, F)$ de F_N à F :

$$d_K(F_N, F) = \sup_x |F_N(x) - F(x)|.$$

La loi limite de $K_N = \sqrt{N} d_K(F_N, F)$ admet pour fonction de répartition :

$$H(x) = \sum_{j=-\infty}^{+\infty} (-1)^j e{-}2j^2k^2, x > 0.$$

La région critique de niveau α du test est $\{K_N > c\}$ où $c = H^{-1}(1-\alpha)$.
Dans la pratique $d_K(F_N, F)$ peut se calculer de la façon suivante :

$$d_K(F_N, F) = \max\left[d^+(F_N, F), d^-(F_N, F)\right]$$

où

$$d^+(F_N, F) = \max_{1 \leq i \leq N} \left[\frac{i}{N} - F\left(x_{(i)}\right)\right] \text{ et } d^-(F_N, F) = \max_{1 \leq i \leq N} \left[F\left(x_{(i)}\right) - \frac{i-1}{N}\right],$$

$x_{(1)}, ..., x_{(N)}$ représentant la statistique d'ordre associée à $x_1, ..., x_N$.

2.1.3 Test de Cramer-von Mises

La statistique qui est ici utilisée s'écrit :

$$\begin{aligned} W_N^2 &= N \int_{\mathbb{R}} \left[F_N(x) - F(x)\right]^2 dF(x) \\ &= \sum_{i=1}^{N} \left[F\left(X_{(i)}\right) - \frac{2i-1}{2N}\right]^2 + \frac{1}{12N}. \end{aligned}$$

Cette statistique admet une loi limite permettant de déterminer la valeur de c définissant la région critique $\left\{W_N^2 > c\right\}$ pour un niveau α donné.

2.2 Méthode des transformations

2.2.1 Inversion de la fonction de répartition

La simulation par inversion de la fonction de répartition constitue la méthode de simulation la plus directe. Elle repose sur le résultat suivant :

Théorème 2.1 *Soit F une fonction de répartition définie sur un intervalle $[a, b]$, de fonction inverse*

$$F^{-1}(u) = \inf\{z \in [a, b] : F(z) \geq u\}.$$

Si U est une variable de loi uniforme sur $[0, 1]$, alors $Z = F^{-1}(U)$ est distribuée suivant F.

Ce résultat découle du fait que $pr(Z \leq z) = pr\left(F^{-1}(U) \leq z\right) = pr(U \leq F(z)) = F(z)$.

Lorsque l'on dispose de l'inverse de la fonction de répartition sous forme analytique, l'inversion de la fonction de répartition est une méthode de simulation exacte (si l'on dispose d'un vrai générateur de lois uniformes). Nous avons reporté dans le tableau 2.2.1 un certain nombre d'exemples courants répondant à ce critère.

Distribution	densité		F^{-1}	forme simplifiée
Beta $\mathcal{B}e(\alpha, 1)$	$\alpha z^{\alpha-1}$	$\alpha > 0, 0 \leq z \leq 1$	$u^{\frac{1}{\alpha}}$	–
Beta $\mathcal{B}e(1, \beta)$	$\beta(1-z)^{\beta-1}$	$\beta > 0, 0 \leq z \leq 1$	$1-(1-u)^{\frac{1}{\beta}}$	$1-u^{\frac{1}{\beta}}$
Exponentielle $\mathcal{E}(\lambda)$	$\frac{1}{\lambda}e^{\frac{-z}{\lambda}}$	$\lambda > 0, z \geq 0$	$-\lambda \ln(1-u)$	$-\lambda \ln u$
Logistique $\mathcal{L}(\alpha, \beta)$	$\frac{e^{-(z-\alpha)/\beta}}{\beta\left[1+e^{-(z-\alpha)/\beta}\right]^2}$	$\beta > 0, z \in \mathbb{R}$	$\alpha + \beta \ln\left(\frac{u}{1-u}\right)$	–
Cauchy $\mathcal{C}(\alpha, \beta)$	$\frac{\beta}{\pi\left[\beta^2+(z-\alpha)^2\right]}$	$\beta > 0, z \in \mathbb{R}$	$\alpha + \beta \tan \pi(u-1/2)$	$\alpha + \frac{\beta}{\tan \pi u}$
Pareto $\mathcal{P}a(\alpha, \beta)$	$\frac{\alpha\beta^\alpha}{z^{\alpha+1}}$	$\alpha > 0, z \geq \beta > 0$	$\beta(1-u)^{-1/\alpha}$	$\beta u^{-1/\alpha}$
Weibull $\mathcal{W}(\alpha, \beta)$	$(\alpha/\beta^\alpha)\, z^{\alpha-1}e^{-(z/\beta)^\alpha}$	$\alpha, \beta > 0, z \geq 0$	$\beta[-\ln(1-u)]^{1/\alpha}$	$\beta[-\ln u]^{1/\alpha}$

Tableau 2.1. Exemples de distributions continues simulables par inversion de la fonction de répartition.

Si l'inversion de la fonction de répartition ne peut être réalisée de façon explicite, seule peut être utilisée une approximation numérique de la solution de l'équation $F(z) = u$. Le calcul d'une telle approximation peut être mené de deux façons différentes :
- en utilisant un algorithme qui résout numériquement $F(z) = u$;
- en utilisant une fonction g approchant F^{-1} et connue de façon analytique.

Résolution numérique de $F(z) = u$

La résolution numérique de l'équation $F(z) = u$ est réalisée au moyen d'un algorithme. Lorsque F est continue, La solution z^* retournée ne peut être qu'approchée : elle dépend de la règle d'arrêt utilisée par l'algorithme mis en œuvre.

Le choix de la règle d'arrêt dépend bien entendu de la distribution que l'on souhaite simuler. A titre d'exemple, on peut considérer le cas d'un algorithme qui retourne la valeur z_N calculée à la N^{ime} itération si $|z_N - z_{N-1}| < \delta$, avec δ petit. Si la valeur z recherchée est très grande une telle règle s'avère inadaptée : on est ici limité par la précision de l'ordinateur. On peut alors envisager une solution alternative prenant l'une des formes suivantes :

$$|z_N - z_{N-1}| < \delta \, |z_N|$$

ou bien

$$|F(z_N) - F(z_{N-1})| < \epsilon.$$

Notons qu'il n'existe pas de règle d'arrêt universelle : celle-ci doit être étudiée au cas par cas. Les trois paragraphes suivants présentes trois algorithmes classiques permettant de résoudre $F(z) = u$.

Algorithme 2.1 – *Algorithme de dichotomie* –

Déterminer un intervalle $[a, b]$ *contenant la solution :* $F(a) \leq u \leq F(b)$.

1. *calculer* $z = \frac{a+b}{2}$
2. *si* $F(z) \leq u$ *alors* $a \leftarrow z$, *sinon* $b \leftarrow z$
3. *si* $b - a < \delta$ *retourner* z, *sinon retourner en* 1.

Algorithme 2.2 – *Algorithme de la sécante* –

Déterminer un intervalle $[a, b]$ *contenant la solution.*

1. *calculer* $z = a + (b - a) \frac{u - F(a)}{F(b) - F(a)}$
2. *si* $F(z) \leq u$ *alors* $a \leftarrow z$, *sinon* $b \leftarrow z$
3. *sib* $- a < \delta$ *retourner* z, *sinon retourner en* 1.

Algorithme 2.3 – *Algorithme de Newton-Raphson* –

Initialiser z

1. $z \leftarrow z - \frac{F(z) - U}{f(z)}$ *où* f *représente la densité associée à la distribution* F.
2. *Si la condition d'arrêt est remplie, retourner* z, *sinon répéter l'étape* 1

Parmi ces trois algorithmes, seul l'algorithme de dichotomie converge dans tous les cas. Si l'équation $F(z) = u$ possède une unique solution, alors l'algorithme de la sécante converge également.

En ce qui concerne l'algorithme de Newton-Raphson, celui-ci converge pour les distributions F concaves ou convexes. Dans le cas (fréquent) où la densité f est unimodale de mode m, alors F est convexe sur $]-\infty, m]$ et concave sur $[m, \infty[$. L'algorithme de Newton-Raphson peut donc être utilisé en prenant m comme valeur initiale.

Notons enfin que si l'algorithme de dichotomie est le plus robuste, les méthodes des sécantes et de Newton-Raphson sont plus rapide. Le lecteur pourra se référer à [20, Ostrowski] pour plus de détails.

Approximation de F^{-1}

Une méthode alternative à la résolution numérique de $F(z) = u$ au moyen d'un algorithme consiste à trouver une fonction g approchant F^{-1} et qui soit connue de façon explicite. L'exemple suivant illustre cette méthode dans le cas de la loi normale.

Exemple 2.1 – *Simulation d'une loi normale* $\mathcal{N}(0, 1)$ *(Hasting, 1955)* –

L'inverse Φ^{-1} *de la fonction de répartition de la loi normale centrée réduite n'est pas connue de façon analytique. La fonction* g *suivante constitue une bonne approximation de* Φ^{-1} *permettant la*

mise en place d'une méthode de simulation par inversion de la fonction de répartition :

$$g(u) = signe\left(u - \frac{1}{2}\right)\left[t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3}\right]$$

où $t = \sqrt{-\ln\left[\min(u, 1-u)\right]}$, *les coefficient* c_i *et* d_i *prenant les valeurs suivantes :*

i	c_i	d_i
0	2.515517	–
1	0.802853	1.432788
2	0.010328	0.189269
3	–	0.001308

L'erreur absolue que l'on réalise en utilisant une telle approximation est strictement inférieure à 0.45×10^{-3}. *Si l'on utilise* $g(u)$ *comme valeur initiale de l'un des algorithme permettant la résolution numérique de de* $F(z) = u$, *cette erreur peut encore être réduite. Sous* GAUSS, *la fonction* $invcdfn(.)$ *proposée par D. Baird emploie une telle méthode, l'algorithme utilisé étant l'algorithme de Newton-Raphson.*

2.2.2 Utilisation de Relations fonctionnelles

L'intérêt de la méthode de simulation par inversion de la fonction de répartition est d'exhiber une relation fonctionnelle simple entre une loi élémentaire (loi uniforme sur $[0,1]$) et une loi plus complexe que l'on souhaite simuler. Une telle approche se généralise grâce au théorème suivant :

Théorème 2.2 *Soit* X *une variable aléatoire de densité* g *sur* $\Delta \subset \mathbb{R}^d$ *et* $\varphi : \Delta \to \mathcal{S} \subset \mathbb{R}^d$ *un* $\mathcal{C}^1$ *- difféomorphisme :* φ^{-1} *est* $\mathcal{C}^1$ *et* $J_{\varphi^{-1}}(x) = det\left(\varphi^{-1}\right)'(x) \neq 0$ *pour tout* $x \in \mathcal{S}$.

Alors, La variable $Y = \varphi(X)$ *a pour densité :*

$$g\left[\varphi^{-1}(x)\right]\left|J_{\varphi^{-1}}(x)\right|, x \in \mathcal{S}.$$

Ce théorème permet d'obtenir de nombreuses distributions sous la forme de une où plusieurs fonctions de variables aléatoires relativement simples.

Exemple 2.2 – *Simulation d'une loi normale* $\mathcal{N}(0,1)$, *algorithme de Box-Muller* –

Ce premier exemple montre comment il est possible de simuler de façon exacte deux lois normales centrées réduites indépendantes à partir de deux lois iid $\mathcal{U}[0,1]$ *: si* U_1 *et* U_2 *sont deux lois uniformes sur* $[0,1]$ *et indépendantes, alors les variables* X_1 *et* X_2 *définies par*

$$\begin{aligned} X_1 &= \sqrt{-2lnU_1}\cos 2\pi U_2 \\ X_2 &= \sqrt{-2lnU_1}\sin 2\pi U_2 \end{aligned}$$

sont indépendantes, de loi $\mathcal{N}(0,1)$.

Pour démontrer ce résultat,il suffit de remarquer que la transformation inverse s'écrit ici :

$$(u_1, u_2) = \left(e^{-\frac{x_1^2+x_2^2}{2}}, \frac{1}{2\pi}\arctan\frac{x_2}{x_1} + k\right) \text{ où } k = \left|\begin{array}{l} 0 \text{ si } (x_1,x_2) \in \mathbb{R}_+ \times \mathbb{R}_+, \\ \frac{1}{2} \text{ si } (x_1,x_2) \in \mathbb{R}_- \times \mathbb{R}, \\ 1 \text{ si } (x_1,x_2) \in \mathbb{R}_+ \times \mathbb{R}_-. \end{array}\right.$$

Un calcul rapide montre que le Jacobien de cette transformation est exactement la densité d'une loi normale centrée réduite en dimension 2.

Exemple 2.3 – *Obtention d'une loi Beta à partir de deux lois Gamma indépendantes* –

Ce deuxième exemple illustre comment il est possible d'obtenir une loi Beta $\mathcal{B}e\,(\alpha,\beta)$ *(*$\alpha > 0$*,* $\beta > 0$*) à partir de deux lois Gamma de paramètres* $(\alpha, 1)$ *et* $(\beta, 1)$*.*

Rappelons dans un premier temps que la densité d'une loi Gamma $\mathcal{G}a(a,b)$ *(*$a > 0, b > 0$*) s'écrit :*

$$\frac{x^{a-1}e^{\frac{-x}{b}}}{\Gamma\,(a)\,b^a}, x \geq 0,$$

celle d'une loi $\mathcal{B}e\,(a,b)$ *(*$a > 0$*,* $b > 0$*) étant :*

$$\frac{\Gamma\,(\alpha+\beta)}{\Gamma\,(\alpha)\,\Gamma\,(\beta)} x^{\alpha-1}\,(1-x)^{\beta-1}\,, x \in [0,1]\,.$$

Dès lors, on peut montrer que si X_1 *et*X_2 *sont deux lois Gamma indépendantes* $\mathcal{G}a(\alpha,1)$ $\mathcal{G}a(\beta,1)$*, alors* $\frac{X_1}{X_1+X_2}$ *suit une loi Beta* $\mathcal{B}e\,(\alpha,\beta)$*. Considérons pour cela la transformation* $y = \frac{x_1}{x_1+x_2}$ *et* $z = x_1 + x_2$*, dont la transformation inverse est* $x_1 = yz$ *et* $x_2 = (1-y)z$*. Le Jacobien de cette transformation s'écrit :*

$$\begin{vmatrix} \frac{\partial x_1}{\partial y} & \frac{\partial x_1}{\partial z} \\ \frac{\partial x_2}{\partial y} & \frac{\partial x_2}{\partial z} \end{vmatrix} = |z|,$$

si bien que la densité f *du couple* $(Y,Z) = \left(\frac{X_1}{X_1+X_2}, X_1 + X_2\right)$ *s'écrit :*

$$\begin{aligned} f\,(y,z) &= \frac{(yz)^{\alpha-1}e^{-yz}}{\Gamma\,(\alpha)} \frac{((1-y)z)^{\beta-1}e^{-(1-y)z}}{\Gamma\,(\beta)} z \\ &= \frac{\Gamma\,(\alpha+\beta)\,y^{\alpha-1}(1-y)^{\beta-1}}{\Gamma\,(\alpha)\,\Gamma\,(\beta)} \frac{z^{\alpha+\beta-1}e^{-z}}{\Gamma\,(\alpha+\beta)}, \end{aligned}$$

ce qui montre le résultat énoncé.

Notons de plus que $X_1 + X_2$ *est indépendante* $\frac{X_1}{X_1+X_2}$ *et qu'il s'agit d'une loi Gamma* $\mathcal{G}a(\alpha+\beta,1)$*.*

De façon générale, la somme de N lois Gamma indépendantes $\mathcal{G}a(a_1,b),...,\mathcal{G}a(a_N,b)$ est une lois Gamma $\mathcal{G}a(\sum a_i, b)$. Cette propriété intéressante permet d'établir les résultats de l'exemple suivant :

Exemple 2.4 – *Utilisation de la loi exponentielle* –

Nous avons vu que si U *suit une loi uniforme sur* $[0,1]$*, alors* $-\lambda \ln U$ *suit une loi exponentielle de paramètre* λ*. En remarquant que la loi exponentielle* $\mathcal{E}\,(\lambda)$ *correspond à la loi* $\mathcal{G}a(1,\lambda)$*, on obtient les résultats suivants :*

– *Si* $U_1,...,U_N$ *sont* N *lois indépendantes, uniformément réparties sur* $[0,1]$*, alors*

$$Y = -2\sum_{i=1}^{N} \ln U_i \sim \chi^2_{2N} = \mathcal{G}a(N,2).$$

– *Si* $U_1,...,U_a,...,U_{a+b}$ *sont* $a+b$ *lois indépendantes uniformément réparties sur* $[0,1]$*, alors*

$$Y = \frac{\sum_{i=1}^{a} \ln U_i}{\sum_{i=1}^{a+b} \ln U_i} \sim \mathcal{B}e(a,b).$$

L'exemple que nous considérons maintenant présente deux méthodes de simulation différente de la loi t de Student.

Exemple 2.5 – *Simulation de la loi t de Student à N degrés de liberté–*

La première façon de simuler une loi de Student à N degrés de liberté repose sur la relation fonctionnelle classique exprimant la loi de Student à partir d'une loi normale et d'un Khi-deux. Si X suit une loi normale $\mathcal{N}(0,1)$ et Y une loi Gamma $\mathcal{G}a\left(\frac{N}{2},2\right)$ (ie un Khi-deux à N degrés de liberté), alors

$$Z = \frac{X}{\sqrt{\frac{Y}{N}}}$$

suit la loi t de Student à N degrés de liberté, i.e. Z a pour densité

$$\frac{\Gamma\left(\frac{N+1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)\sqrt{\pi N}}\left(1+\frac{z^2}{N}\right)^{-\frac{N+1}{2}}$$

La demonstration de ce résultat est directe en utilisant le théorème 2.2 page 15.

La simulation d'une loi de Student à N degrés de liberté peut également être réalisée en 1 ligne, simplement à partir de deux lois uniformes indépendantes. En effet, si U et V sont iid $\mathcal{U}(0,1)$, alors

$$X = \sqrt{N\left(U^{-\frac{2}{N}}-1\right)}cos2\pi V \sim t_N$$

Un tel résultat peut bien entendu être démontrer à partir du théorème 2.2 page 15. Le lecteur intéressé par la façon dont on peut établir une telle relation pourra se reporter à [7, Devroye]

Exemple 2.6 – *Simulation d'une loi Beta symétrique –*

La simulation d'une loi Beta $\mathcal{B}e(a,a)$ ($a > \frac{1}{2}$) peut être réalisée de la façon suivante (Ulrich, 1984) :

Si U et V sont deux lois uniformes sur $[0,1]$, indépendantes, alors

$$X = \frac{1}{2}+\frac{1}{2}\sin 2\pi V\sqrt{1-U^{\frac{2}{2a-1}}} \sim \mathcal{B}e(a,a)$$

Notons que cette relation ne permet pas de simuler de loi Beta symétrique de paramètre inférieur à $\frac{1}{2}$. On peut remédier à ce problème en exploitant le résultat démontré par Best (1978) :

$$\mathcal{B}e(a,a) \overset{loi}{=} \frac{1}{2}\left(1+\frac{t_{2a}}{\sqrt{2a+t_{2a}^2}}\right)$$

Dès lors, compte tenu de la deuxième méthode de simulation de la loi de Student que nous avons présenté précédemment, si U et V sont iid $\mathcal{U}(0,1)$ et si S est un signe aléatoire équi-distribué, alors

$$X = \frac{1}{2}+\frac{1}{2}\frac{S}{\sqrt{1+\frac{1}{\left(U^{-\frac{1}{a}}-1\right)cos^2 2\pi V}}} \sim \mathcal{B}e(a,a)$$

2.3 Le cas des distributions discrètes

La distribution d'une variable aléatoire discrète est déterminée par la donnée d'une suite p_0,...,p_k,... telle que :

$$pr(X=i) = p_i$$

Si le support de X est fini, cette distribution est connue sous forme de table. Sinon, il peut s'agir d'une expression analytique (par exemple, $p_i = \frac{\lambda^i e^{-\lambda}}{i!}$ dans le cas de la loi de Poisson $\mathcal{P}(\lambda)$).

2.3.1 Méthode d'inversion, le cas général

La méthode de simulation par inversion de la fonction répartition que nous avons présentée en début de partie s'applique également dans le cas des distributions discrètes.

Si U est uniformément distribué sur $[0,1]$, alors si l'on définit X par

$$F(X-1) = \sum_{i<X} p_i < U \leq \sum_{i \leq X} p_i,$$

X est distribué suivant F : $pr(X = i) = pr(X \leq i) - pr(X \leq i-1) = F(i) - F(i-1) = p_i$. L'algorithme suivant constitue l'algorithme le plus simple permettant d'inverser la distribution F.

Algorithme 2.4 – *Algorithme d'inversion séquentiel* –

1. *Poser* $X = 0$, $S = p_0$, *et simuler* $U \sim \mathcal{U}(0,1)$
2. *Tant que* $S < U, X \leftarrow X+1$ *et* $S \leftarrow S + p_X$
3. *retourner* X

Exemple 2.7 – *Simulation d'une loi de Poisson* $\mathcal{P}(\lambda)$ –

Dans le cas de la loi de Poisson de paramètre λ,

$$p_0 = e^{-\lambda} \text{ et } p_{i+1} = \frac{\lambda}{i+1} p_i.$$

L'algorithme précédent se réécrit donc :

1. *Initialiser* $X \leftarrow 0$, $P \leftarrow e^{-\lambda}$ *et* $S \leftarrow P$;
2. *Simuler* $U \sim \mathcal{U}(0,1)$
3. *Tant que* $S < U, X \leftarrow X+1$, $P \leftarrow \frac{\lambda}{X} P$ *et* $S \leftarrow S + P$
4. *retourner* X

Notons que le temps de calcul associé à cet algorithme croît de façon linéaire en l'espérance de la variable à simuler. En effet, si l'on note N le nombre de comparaisons nécessaires au cours de l'algorithme 2.4,

$$pr(N = i) = pr(X = i-1) = p_{i-1},$$

si bien que $\mathbb{E}N = \mathbb{E}X + 1$.

On voit ici que cet algorithme n'est pas adapté à la simulation de lois dont l'espérance est élevée (c'est par exemple le cas de la loi de Poisson lorsque λ est grand).

2.3.2 Inversion à partir de la distribution d'une loi continue

Une méthode plus efficace que la précédente peut être mise en place lorsqu'on dispose d'une fonction de distribution continue sur $\mathbb{R}_+$ coïncidant sur $\mathbb{N}$ avec la distribution discrète F que l'on souhaite simuler :

$$G(i+1) = F(i), \forall i, G(0) = 0.$$

L'algorithme suivant peut alors être utilisé.

Algorithme 2.5 – *Algorithme d'inversion à partir d'une distribution continue tronquée* –

1. *simuler* $U \sim \mathcal{U}(0,1)$
2. *retourner* $X = \lfloor G^{-1}(U) \rfloor$

Il s'agit bien évidemment d'une méthode extrêmement rapide lorsque G^{-1} est connue de façon explicite. Le résultat découle simplement du fait que

$$pr\left(\left\lfloor G^{-1}(U)\right\rfloor \leq i\right) = pr(G^{-1}(U) < i+1) = G(i+1) = F(i).$$

Exemple 2.8 – *Simulation d'une loi géométrique de paramètre p* –

La loi géométrique est définie par $pr\,(X = i) = p(1-p)^{i-1}, i \geq 1$. *Définissons alors* $G(x) = 1 - e^{-\lambda x}$ *avec* $p = 1 - e^{-\lambda}$. *La fonction G vérifie :*

$$G(i+1) - G(i) = e^{-\lambda i}\left(1 - e^{-\lambda}\right) = p(1-p)^i,$$

si bien que $X = \left\lceil \frac{lnU}{ln(1-p)} \right\rceil$, *suit une loi géométrique de paramètre p si* $U \sim \mathcal{U}(0,1)$.

Exemple 2.9 – *Simulation d'une loi uniforme de paramètre sur* $\{1, ..., K\}$ –

La loi uniforme est définie par $pr\,(X = i) = \frac{1}{K}$. *En considérant*

$$G(x) = \frac{x-1}{K}, 1 \leq x \leq K+1,$$

on voit $X = \lfloor Ku + 1 \rfloor$, *est uniformément distribué sur* $\{1, ..., K\}$.

2.3.3 La méthode de la table guide

Cette méthode consiste à accélérer l'algorithme de recherche séquentielle 2.4 page précédente. Comme nous l'avons vu, le défaut de cet algorithme est qu'il nécessite en moyenne $\mathbb{E}X$ comparaisons au cours de l'étape de recherche avant de retourner une valeur. La méthode proposée ici cherche à réduire le nombre de comparaisons. Pour cela, il est nécessaire de déterminer un candidat plus pertinent que la modalité 0 pour initialiser la phase de recherche.

Notons F la distribution de la variable X à simuler et U un élément de [0,1]. Le candidat retenu par la méthode de la table guide est la première modalité x de la variable aléatoire à simuler dont on soit sûr qu'elle vérifie : $F(x) > U$

Soit N+1 le nombre de modalité de X. A chacun des intervalles $\left[\frac{i}{N+1}, \frac{i+1}{N+1}\right[$ $(0 \leq i \leq N)$, on associe la plus grande modalité g_i telle que F(x) $< \frac{i+1}{N+1}$, i.e. :

$$F(g_i) < \frac{i+1}{N+1} \leq F(g_i + 1)$$

Ainsi si l'on sait que $u \in \left[\frac{i}{N+1}, \frac{i+1}{N+1}\right[$, la modalité à retenir pour initialiser l'algorithme est $g_i + 1$.

Pour une valeur de u donnée, l'indice i_u vérifiant $\frac{i}{N+1} \leq u < \frac{i+1}{N+1}$ s'écrit

$$i_u = \lfloor (N+1)u \rfloor$$

L'algorithme reposant sur cette analyse s'écrit de la façon suivante :

Algorithme 2.6 – *Méthode de la table guide* –

Une fois la table des $g_i, i \in \{0, ..., N\}$ *initialisée,*

1. *Simuler* $U \sim \mathcal{U}(0,1)$
2. *Poser* $X = \lfloor (N+1)u \rfloor$
3. $X \leftarrow g_X + 1$

4. *Tant que* $F(X-1) > U, X \leftarrow X-1$
5. *Retourner* X

Le nombre N de comparaisons nécessaires au cours de l'algorithme est ici majoré en moyenne par 2. En effet,

$$\mathbb{E}N \leq 1 + \sum_{i=0}^{N} pr\left(\lfloor (N+1)u \rfloor = i\right) \times \{\text{nombre de modalités situées dans l'intervalle } i\} = 2.$$

2.3.4 Le cas particulier de la loi de Poisson

Nous présentons ici une méthode de simulation de la loi de Poisson exploitant le lien qui existe entre le processus de Poisson d'intensité λ et la loi exponentielle $\mathcal{E}\left(\frac{1}{\lambda}\right)$.

Définition 2.1 *Soit* $(T_i)_{i\geq 1}$ *une suite de variables aléatoire iid de loi* $\mathcal{E}\left(\frac{1}{\lambda}\right)$ *(i.e. de densité* $\lambda e^{-\lambda t}\mathbf{1}_{\{t\geq 0\}}$*), et posons* $\tau_n = \sum_{i=1}^{n}$.
On appelle processus de Poisson d'intensité λ *le processus* N_t *défini par*

$$N_t = \sum_{n\geq 1} n\mathbf{1}_{\{\tau_n \leq t < \tau_{n+1}\}}.$$

Proposition 2.1 *Si* $(N_t)_{t\geq 0}$ *est un processus de Poisson d'intensité* λ*, alors* N_t *suit une loi de Poisson de paramètre* λt *:*

$$pr\left(N_t = n\right) = e^{-\lambda t}\frac{(\lambda t)^n}{n!}.$$

La démonstration de ce résultat s'effectue simplement en remarquant que

$$pr(N_t = n) = pr(\tau_n \leq t) - pr(\tau_{n+1} \leq t),$$

et τ_n suit une loi Gamma $\mathcal{G}a\left(n, \frac{1}{\lambda}\right)$ (cf. exemple 2.4 page 16).

Compte tenu de ce résultat, N_1 suit une loi de Poisson $\mathcal{P}(\lambda)$, si bien que la simulation d'une loi de Poisson X de paramètre λ peut être réalisée en simulant N_1 lois exponentielles $\mathcal{E}\left(\frac{1}{\lambda}\right)$ (notées T_i), où N_1 vérifie :

$$T_1 + ...T_{N_1} \leq 1 < T_1 + ...T_{N_1}.$$

Il suffit ensuite de retourner $X = N_1$.

On sait de plus que lorsque $U \sim \mathcal{U}(0,1)$, $-\frac{1}{\lambda}lnU \sim \mathcal{E}\left(\frac{1}{\lambda}\right)$, si bien que N_1 se caractérise également par la relation suivante :

$$U_1U_2...U_{N_1+1} < e^{-\lambda} \leq U_1U_2...U_{N_1}, U_i \sim \mathcal{U}(0,1)$$

On peut donc simuler une loi de Poisson à partir de l'algorithme suivant :

Algorithme 2.7 – *Simulation d'une loi de Poisson* $\mathcal{P}(\lambda)$ –
Poser $a = e^{-\lambda}$.

1. $N \leftarrow 0$, $P \leftarrow U$ *où* $U \sim \mathcal{U}(0,1)$.
2. *Tant que* $P < a$, $P \leftarrow P * U$ *où* $U \sim \mathcal{U}(0,1)$ *et* $N \leftarrow N+1$.
3. *Retourner* N.

2.4 La méthode des mélanges discrets

Supposons que le densité de la variable X que l'on cherche à simuler puisse s'écrire sous la forme :

$$f(x) = \sum_{i=1}^{r} p_i f_i(x)$$

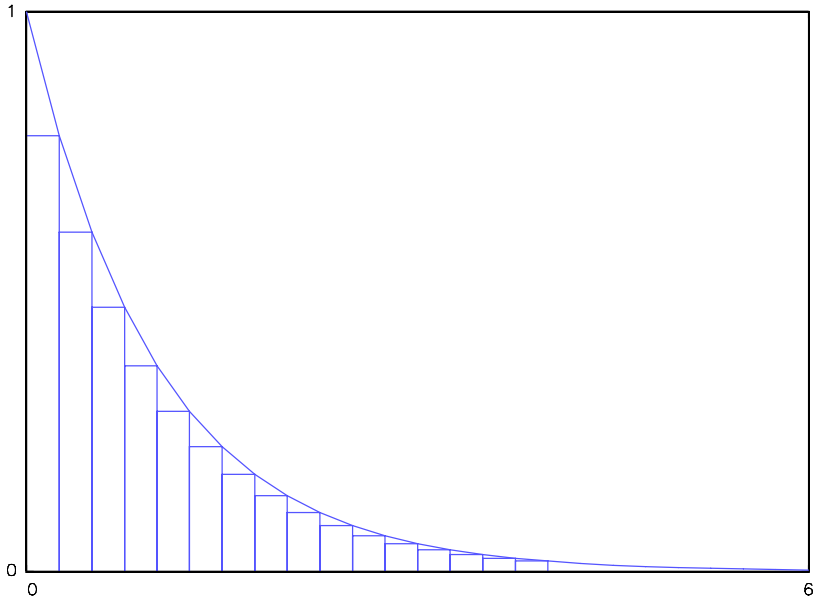
où $p_1, ..., p_r$ représentent une distribution discrète ($0 < p_i < 1$ et $\sum p_i = 1$) et f_i sont des densités définies sur $[a_i, b_i]$. Alors X peut être simulé à partir de l'algorithme suivant :

Algorithme 2.8 – *Méthode des mélanges* –

- *Simuler* $I \sim \{p_1, ..., p_r\}$
- *Simuler* X *suivant* $f_I(x), x \in [a_i, b_i]$
- *retourner* X.

L'exemple que nous considérons ici montre comment il est possible de simuler une loi exponentielle sans passer par l'évaluation d'un logarithme (ce qui s'avérait autrefois coûteux en temps de calcul, c'est aujourd'hui moins le cas).

Exemple 2.10 – *Simulation de la loi exponentielle, "The rectangular-wedge-tail method"* – ,



Graphique 2.1. Simulation de la loi exponentielle : "The rectangular-wedge-tail method"

La méthode proposée par à McLaren et al. (1964) repose sur la decomposition suivante de la densité $\mathcal{E}(1)$ *:*

$$f(x) = e^{-x} = \sum_{i=1}^{2k+1} p_i f_i(x),$$

les coefficients étant définis par :

- $i = 1, ..., k$: $p_i = \lambda e^{-\lambda i}$
  $f_i(x) = \frac{1}{\lambda}, \quad (i-1)\lambda \leq x \leq i\lambda$;

- $i = k+1, ..., 2k$: $p_i = \left(e^{\lambda} - 1 - \lambda\right) e^{-\lambda(i-k)}$
  $f_i(x) = \frac{e^{-x+\lambda(i-k)}-1)}{e^{\lambda}-1-\lambda}$, $(i-k-1)\lambda \leq x \leq (i-k)\lambda$;
- $i = 2k+1$: $p_{2k+1} = e^{-\lambda k}$
  $f_{2k+1}(x) = e^{-x+\lambda(k)}, x \geq \lambda k$.

Ainsi, si les composantes 1 *à* k *sont tirés, on simule une loi uniforme.*

Les composantes $k+1$ *à* $2k$ *peuvent être simulées par la méthode d'acceptation-rejet présentée dans la section suivante.*

En ce qui concerne la composante $2k+1$*, il convient de remarquer que*

$$pr(X > x | X > k\lambda) = 1 - e^{-(x-k\lambda)}.$$

Par conséquent, $(X - k\lambda | X > k\lambda)$ *suit une loi* $\mathcal{E}(1)$*. La simulation suivant* f_{2k+1} *peut donc être réalisée en tirant dans la loi* $\{p_1, ..., p_{2k+1}\}$ *jusqu'à ce que l'indice* J *tiré soit inférieur ou égal à* $2k$*. On retourne* $kI\lambda + X, X \sim f_J$*, où* I *représente le nombre de tirages nécessaires pour que* $J \leq 2k$*.*

2.5 Simulation par acceptation-rejet

La simulation par acceptation rejet constitue une méthode alternative à la méthode des transformation lorsque l'on ne réussit pas à écrire la loi à simuler comme une transformée de variables. Ce type de méthode nécessite la connaissance de la densité à de la loi à simuler une constante multiplicative près, et repose sur l'utilisation d'une loi instrumentale que l'on sait déjà simuler.

2.5.1 Le cas général

Nous commençons par présenter les théorèmes sur lesquelles se fonde la simulation par acceptation-rejet.

Théorème 2.3 *Soit* $U \sim \mathcal{U}(0,1)$ *et* X *une variable aléatoire indépendante de* U *de densité* f *sur* $\mathbb{R}^d$*.*

Alors $(X, cUf(X))$ *est uniformément distribué sur* $A = \left\{(x,v) : x \in \mathbb{R}^d, 0 \leq v \leq cf(x)\right\}$*,* c *étant une constante positive arbitraire.*

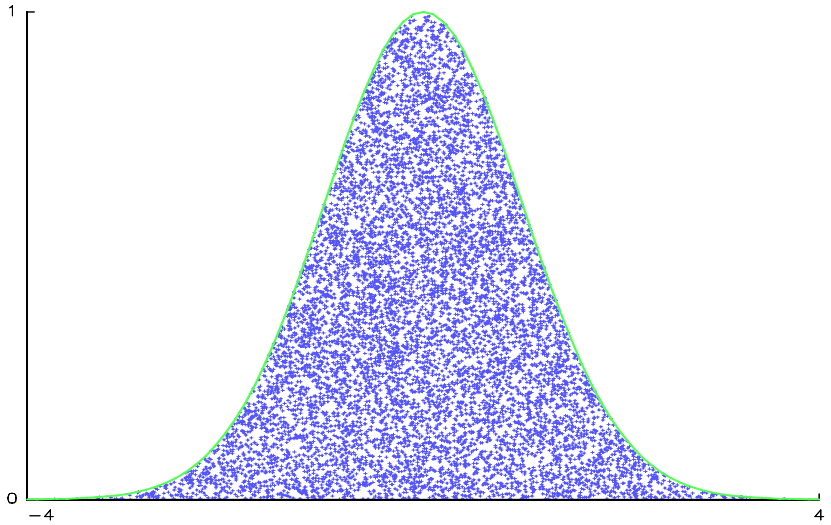
réciproquement, si (X, V) *est uniformément distribué sur* A*, alors* X *est distribué suivant* f*.*

Le graphique 2.2 page suivante illustre ce théorème 2.3 dans le cas où $X \sim \mathcal{N}(0,1)$.

Théorème 2.4 *Soit* $X_1, X_2, ...$ *une famille de variables aléatoires iid à valeurs dans* $\mathbb{R}^d$ *et* $A \subset \mathbb{R}^d$*.*
Si $Y = X_I$ *où* $I = \min\{j : X_j \subset A\}$*, alors pour tout* B *mesurable :*

$$pr(Y \in B) = \frac{pr\left(x_1 \in A \cap B\right)}{pr\left(X_1 \in A\right)}$$

En particulier si X_1 *et uniformément distribuée sur* $A_0 \supset A$ *alors* Y *est uniformément distribuée sur* A*.*



Graphique 2.2. Distribution uniforme sur $A = \left\{(x, v) : x \in \mathbb{R}^d, 0 \leq v \leq e^{-\frac{x^2}{2}}\right\}$

Représentation de 1000 couples $\left(X, Ue^{-\frac{X^2}{2}}\right)$ avec $X \sim \mathcal{N}(0, 1)$.

Pour démontrer ce résultat, il suffit de remarquer que :

$$\begin{aligned} pr(Y \in B) &= \sum_{i=1}^{\infty} pr\left(X_1 \notin A, X_{i-1} \notin A, X_i \in A \cap B\right) \\ &= \sum_{i=1}^{\infty} (1-p)^{i-1} pr\left(X_1 \in A \cap B\right), \text{ avec } p = pr(X_1 \in A) \\ &= \frac{pr\left(X_1 \in A \cap B\right)}{p}. \end{aligned}$$

L'algorithme général de simulation par acceptation-rejet découle directement des deux théorèmes qui viennent d'être énoncés.

Algorithme 2.9 – *Méthode d'acceptation-rejet* –

Soit f la densité de la variable aléatoire que l'on souhaite simuler. On suppose connue une densité g et une constante c telles que : $\forall x, f(x) \leq cg(x)$.

1. *Simuler $U \sim \mathcal{U}(0, 1)$ et $X \sim g$ de façon indépendante.*
2. *si $cUg(X) \leq f(X)$, retourner X, sinon reprendre l'étape 1.*

D'après la première partie du théorème 2.3, le couple $(X, cg(X))$ est uniformément distribué sous la courbe $c.g$. Le théorème 2.4 montre alors que la variable $(X, cg(X))$ retournée par l'algorithme est uniformément distribuée sous la courbe f et par conséquent, compte tenu de la deuxième partie du théorème 2.3, X est distribuée suivant f.

Nombre moyen d'itérations.

Notons N le nombre d'itérations nécessaires pour retourner une valeur au cours de l'algorithme d'acceptation-rejet.

$$pr(N = i) = (1-p)^{i-1}p \text{ et } pr(N \geq i) = (1-p)^{i-1}$$

où

$$\begin{aligned} p &= pr(cUg(X) \leq f(X)) \\ &= \int pr\left(U \leq \frac{f(x)}{cg(x)}\right) g(x)dx \\ &= \int \frac{f(x)}{c} dx \\ &= \frac{1}{c} \end{aligned}$$

Par conséquent $\mathbb{E}N = \frac{1}{p} = c$: la probabilité d'accepter la variable simulée suivant g est égale à l'inverse de c. Il convient donc de choisir c le plus grand possible.

Exemple 2.11 – *Simulation d'une loi normale à partir de la densité de Laplace* –

La densité de la loi de Laplace s'écrit :

$$g(x) = \frac{1}{2}e^{-|x|}, x \in \mathbb{R}$$

Si on note f la densité de la loi normale $\mathcal{N}(0,1)$, on peut montrer que

$$f(x) \leq cg(x) \text{ avec } c = \sqrt{\frac{2e}{\pi}}$$

On peut donc mettre en place l'algorithme suivant :

Algorithme 2.10

1. *Simuler une loi exponentielle $\mathcal{E}(1)$ et deux loi U et V iid $\mathcal{U}(0,1)$. Si $U < \frac{1}{2}$ $X \leftarrow -X$ (X est maintenant distribué suivant la loi de Laplace)*
2. *si $V\frac{1}{\sqrt{2\pi}}e^{\frac{1}{2}-|X|} \leq \frac{1}{\sqrt{2\pi}}e^{-\frac{X^2}{2}}$, accepter X, sinon retourner en 1.*

Exemple 2.12 – *Le "Wedge generator"* –

On se propose ici de simuler la loi de densité

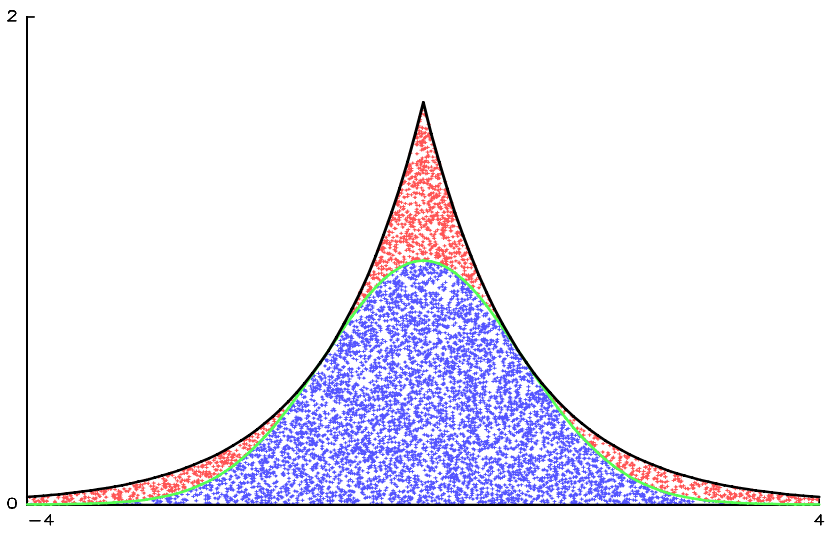
$$f(x) = \frac{e^{\lambda - x} - 1}{e^{\lambda} - 1 - \lambda}$$

Il s'agit de la fonction intervenant dans la méthode de simulation de la loi exponentielle présentée à la section 2.4 page 21.

La méthode d'acceptation-rejet peut s'appliquer ici en remarquant que

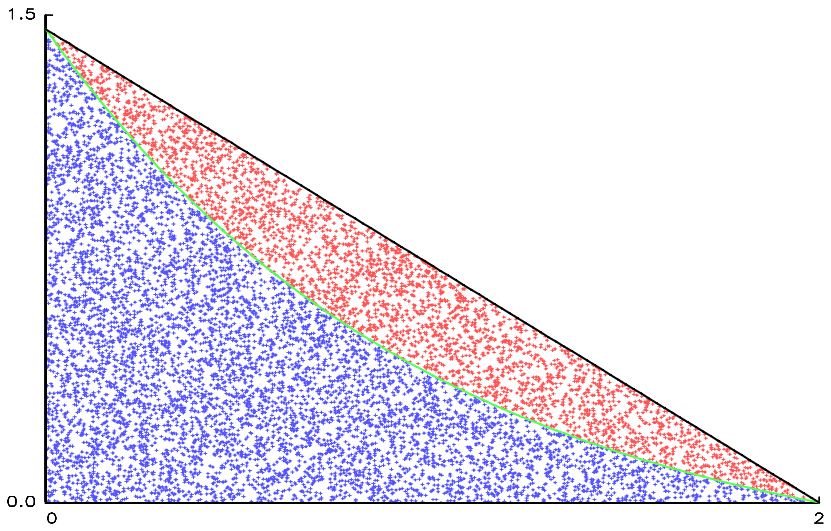
$$f(x) \leq d(x) = \frac{\lambda - x}{\lambda} \frac{e^{\lambda} - 1}{e^{\lambda} - \lambda - 1}$$

Dès lors, pour obtenir une variable aléatoire distribuée suivant f, il suffit de simuler un couple de coordonnées positives et uniformément répartie sous la droite d'équation $y = d(x)$.



Graphique 2.3. Simulation d'une loi normale à partir de la loi de Laplace

L'abscisse des points acceptés (en bleu) par l'algorithme est distribuée suivant la loi $\mathcal{N}(0,1)$



Graphique 2.4. Simulation de la densité $f(x) = \frac{e^{\lambda-x}-1}{e^{\lambda}-\lambda-1}, 0 \leq x \leq \lambda, \lambda = 2$.

La simulation est réalisée à partir d'un couple de variables positives uniformément réparties sous la droite

$$y = \frac{\lambda - x}{\lambda} \frac{e^{\lambda} - 1}{e^{\lambda} - \lambda - 1}.$$

L'abscisse des points acceptés (en bleu) est distribuée suivant la loi cible.

2.5.2 La méthode du "squeeze"

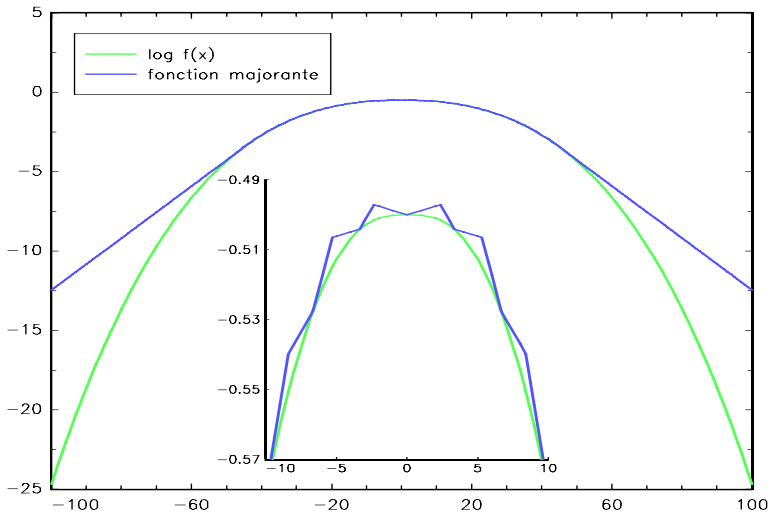
La méthode du "squeeze" permet de raffiner l'algorithme 2.9 page 23. Cette méthode est intéressante lorsque la densité f de la loi à simuler est longue à calculer. Pour l'appliquer, il est nécessaire

de connaître deux fonctions h_1 et h_2 faciles à évaluer et encadrant f : $h_1 \leq f \leq h_2$. L'algorithme que l'on met en place est alors le suivant :

Algorithme 2.11 *Étant données une loi instrumentale g, une constante c telle que $f \leq cg$, et deux fonctions h_1 et h_2 encadrant f,*

1. *Simuler $U \sim \mathcal{U}(0,1)$ et $X \sim g$ de façon indépendante, et pose $W = cg(X)$.*
2. *Si $W \leq h_1(X)$, accepter X ;*
   Sinon, si $W \leq h_2(X)$ alors – si $W \leq f(X)$, accepter X ;
   – sinon retourner en 1.

2.5.3 Le cas des densité log-concaves : La méthode des cordes



Graphique 2.5. Enveloppe supérieure du log d'une densité log-concave

Si la densité f de la loi que l'on l'on souhaite simuler est log-concave, il est possible de définir une fonction majorante de $\log f$ affine par morceau à partir des cordes de $\log f$. Notons l_N une telle fonction et $\mathcal{S}_N$ l'ensemble des points $x_0, x_1, ..., x_N, x_{N+1}$ tels que l_N soit affine sur $[x_i, x_{i+1}]$, avec $x_0 = -\infty$ et $x_{N+1} = +\infty$. Soit alors $\{(\lambda_i, \mu_i), 0 \leq i \leq N\}$ l'ensemble des couples tels que

$$l_N(x) = \mu_i x + \lambda_i \text{ si } x_i \leq x \leq x_{i+1}$$

Posons

$$\varpi = \frac{e^{\lambda_0 + \mu_0 x_1}}{\mu_0} + \sum_{i=1}^{N-1} e^{\lambda_i} \frac{e^{\mu_i x_{i+1}} - e^{\mu_i x_i}}{\mu_i} - \frac{e^{\lambda_N + \mu_N x_N}}{\mu_N}$$

La fonction de répartition L_N associée à l_N s'écrit, en fonction des paramètres que l'on vient d'introduire :

$$L_N(x) = p_k + e^{\lambda_k} \frac{e^{\mu_k x} - e^{\mu_k x_k}}{\varpi \mu_k} \text{ si } x_k < x \leq x_{k+1}, 0 \leq k \leq N$$

où

$$p_k = \sum_{i=0}^{k-1} e^{\lambda_i} \frac{e^{\mu_i x_{i+1}} - e^{\mu_i x_i}}{\varpi \mu_i}.$$

L'algorithme suivant exploite la fonction de répartition L_N pour simuler des variables distribuées suivant une densité f log-concave :

Algorithme 2.12 – *Méthode des cordes* –

1. *Générer deux lois uniformes indépendantes U_1 et U_2 sur $[0,1]$.*
2. *Déterminer $k \in \{0,\ldots,N-1\}$ tel que $p_k < U_2 \leq p_{k+1}$ et calculer $X = L_N^{-1}(U_2) = \frac{1}{\mu_k}\ln\left[\varpi\mu_k e^{-\lambda_k}(U_2 - p_k) + e^{\mu_k x_k}\right]$*
3. *Si*
$$U_1 \leq \frac{f(X)}{l_N(X)},$$
accepter X, sinon retourner en 1.

Une caractéristique intéressante de cet algorithme est qu'il permet la mise à jour de l'ensemble de points $\mathcal{S}_N$ à chaque fois que $f(x)$ est calculé (étape 3 de l'algorithme) : la fonction majorante l_N devient alors plus précise à mesure que le nombre de simulations augmente, ce qui réduit le nombre de rejets.

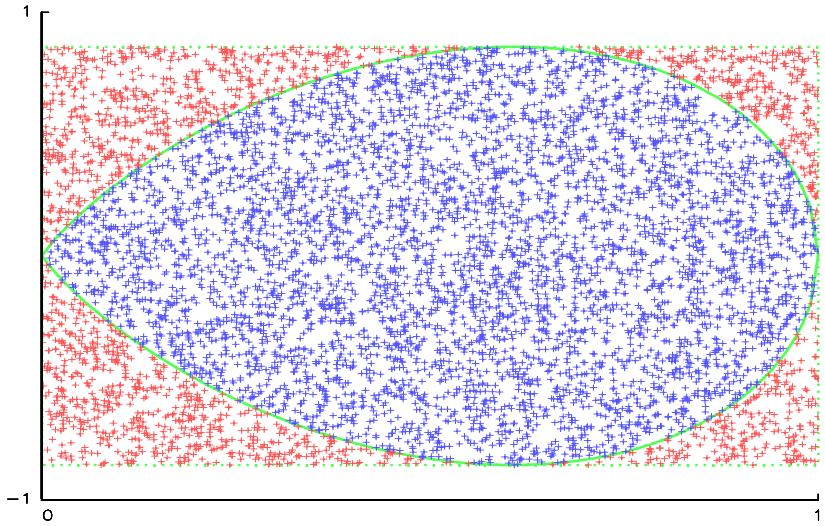
De plus il peut être amélioré s'il est plus facile de calculer la fonction minorante construite à partir des cordes que la fonction f elle même.

2.5.4 La méthode du rapport d'uniformes.

La méthode du rapport d'uniforme repose sur le résultat suivant :

Théorème 2.5 *Soit $\mathcal{D} = \left\{(x,y), 0 \leq x \leq \sqrt{f\left(\frac{y}{x}\right)}\right\}$.*

Si (X,Y) est uniformément distribué sur $\mathcal{D}$, alors $\frac{Y}{X}$ a pour densité $\frac{1}{c}f$ où $c = 2Aire(\mathcal{D})$.



Graphique 2.6. Simulation d'une loi normale par la méthode du rapport d'uniforme. représentation du domaine d'acceptation

Le ratio $\frac{Y}{X}$ des points acceptés (en bleu) est distribué suivant la loi $\mathcal{N}(0,1)$

Nous avons vu précédemment comment simuler une variable uniformément répartie sur le domaine $\mathcal{D}$: il suffit d'appliquer le principe de rejet en incluant $\mathcal{D}$ dans un domaine $\mathcal{P}$ plus simple,

tel qu'un rectangle. En général, on définit $\mathcal{P}$ de la façon suivante :

$$\mathcal{P} = \{(x, y), 0 \leq x \leq x^*, y_* \leq y \leq y^*\}$$

avec

$$\begin{cases} x^* = & \sup_z \sqrt{f(z)} \\ y_* = & \inf_z z\sqrt{f(z)} \\ y^* = & \sup_z z\sqrt{f(z)} \end{cases},$$

On peut alors mettre en place l'algorithme suivant :

Algorithme 2.13 *Une fois initialisées les constantes x^*, y_*, y^**

1. *simuler de façon indépendante $U_1 \sim \mathcal{U}(0, x^*)$ et $U_2 \sim \mathcal{U}(y_*, y^*)$. $Z \leftarrow \frac{U_2}{U_1}$*
2. *Si $U_1^2 \leq f(Z)$, accepter Z, sinon reprendre l'étape 1.*

Exemple 2.13 – *Simulation de la loi normale $\mathcal{N}(0, 1)$* –

Dans le cas de la loi normale,

- $f(z) = e^{-\frac{z^2}{2}}$,
- $x^* = 1$, $y^* = \sqrt{2}e$ *et* $y_* = -\sqrt{2}e$
- *L'aire du domaine $\mathcal{D}$ vaut* $\frac{4}{\sqrt{\pi e}}$
- *La condition d'acceptation s'écrit* $z^2 \leq -4ln(x)$

3

Méthodes de Monte Carlo appliquées au pricing d'option : calcul d'intégrales, accélération de la convergence

Le cadre d'analyse est posé de la façon suivante. On considère un espace probabilisé $(\Omega, \mathcal{F}, \mathbb{Q})$ et on se donne un variable aléatoire X à valeurs dans un espace $\mathcal{S}$, de loi $\mu(dx)$. On souhaite estimer numériquement, à partir d'une méthode de simulation de Monte Carlo, le paramètre d'intérêt

$$I_g = \int_{\mathcal{S}} g(x)\,\mu(dx) = \mathbb{E}^{\mathbb{Q}}[g(X)] \tag{3.1}$$

où g représente une fonction μ-intégrable. La méthode de Monte Carlo standard consiste à réaliser N tirages aléatoires indépendants $(x_1, \ldots, x_N)$ suivant la loi μ et à considérer

$$c_g = \frac{1}{N}\sum_{k=1}^{N} g(x_k) \tag{3.2}$$

comme estimateur du paramètre I_g. Il s'agit ici d'une application directe de la loi forte des grands nombres qui, compte tenu des hypothèses que l'on vient de formuler, assure que

$$\lim_{N\to\infty} c_g = I_g \ p.s.$$

L'objet de cette partie est de présenter les méthodes usuelles qui permettent de construire des estimateurs de I_g convergeant plus rapidement que l'estimateur standard c_g. Dans le cas des méthodes de Monte Carlo, le contrôle de la convergence se traduit en terme de variance : l'objet des méthodes d'accélération est de déterminer des estimateurs de I_g dont la variance est inférieure à $var(c_g)$. Plus cette variance sera faible et plus l'intervalle de confiance centré sur l'estimateur sera petit (cf. théorème central limite).

Utilisation d'une variable de contrôle, échantillonnage d'importance , stratification et utilisation de variables antithétiques sont les quatre méthodes de réduction de la variance exposées dans cette partie. Dans chacun des cas, nous présentons un exemple d'application à la finance afin de montrer comment ces méthodes peuvent être utiliser pour l'évaluation de produits dérivés.

3.1 Contrôle de la convergence

Considérons l'estimateur standard c_g du paramètre d'intérêt I_g présenté en introduction. En appliquant la loi forte des grands nombres, on a :

$$\lim_{N\to\infty} c_g = I_g \ p.s.$$

Comme dans le cas des méthodes d'intégration déterministes, l'erreur d'approximation liée à c_g décroît lorsque N augmente. Néanmoins, l'erreur ne s'exprime pas ici en terme d'approximation mathématique mais suivant des considérations statistiques liées à l'aléa de l'échantillonnage. En particulier, comme nous l'avons déjà noté, la convergence de c_g vers le paramètre d'intérêt I_g est une convergence presque sure, ce qui signifie que

$$\mathbb{Q}\left(\lim_{N\to\infty} c_g = I_g\right) = 1.$$

Si cette propriété est bien évidemment souhaitable, elle ne permet pas d'évaluer l'erreur commise lorsque N est fini.

Nous présentons ici la méthode d'évaluation de l'erreur communément utilisée en pratique pour résoudre ce problème. Elle repose sur la construction d'un intervalle de confiance. De plus, nous explicitons les approximations réalisées afin de mettre en évidence les limites d'un tel intervalle.

L'intervalle $IC(\lambda)$ que l'on souhaite expliciter est centré sur l'estimateur c_g, et tel que le paramètre d'intérêt I_g appartienne à $IC(\lambda)$ avec probabilité $1-\lambda$:

$$\mathbb{Q}\left(I_g \in IC(\lambda)\right) = 1-\lambda$$

Lorsque le moment d'ordre 2 de $g(X)$ est fini, le théorème central limite assure que :

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} g(x_i) - I_g\right) \overset{loi}{\to} \mathcal{N}\left(0, var\left[g(X)\right]\right).$$

Ainsi, pour N suffisamment grand, on peut supposer que l'estimateur c_g est distribué suivant une loi $\mathcal{N}\left(I_g, \frac{var[g(X)]}{N}\right)$. Cette première approximation permet d'écrire $IC(\lambda)$ sous la forme suivante :

$$IC(\lambda) = \left[c_g - \sqrt{\frac{var[g(X)]}{N}}\Phi^{-1}\left(1-\frac{\lambda}{2}\right), c_g + \sqrt{\frac{var[g(X)]}{N}}\Phi^{-1}\left(1-\frac{\lambda}{2}\right)\right] \tag{3.3}$$

où Φ représente la fonction de répartition de la loi normale centrée réduite. Il est important de remarquer que l'approximation réalisée induit ici une première erreur sur la définition de l'intervalle de confiance, erreur qui ne peut être contrôlée : si les queues de distribution de la véritable loi de c_g sont plus épaisses que pour la loi normale, alors l'utilisation de l'intervalle $IC(\lambda)$ défini par la relation (3.3) sur-estime l'efficacité de l'estimateur c_g et, inversement, l'efficacité de c_g est sous-estimée si les queues de distribution sont plus fines.

Par ailleurs, le paramètre $var\left[g(X)\right]$ est *a priori* inconnu. Néanmoins, on peut l'estimer à partir de l'échantillon des variables x_i simulées pour construire c_g. Si l'on note v_g l'estimateur de la variance de $g(X)$ construit à partir des réalisations x_i, on a :

$$v_g = \frac{1}{N}\sum_{i=1}^{N} g(x_i)^2 - \left(\frac{1}{N}\sum_{i=1}^{N} g(x_i)\right)^2.$$

L'intervalle de confiance s'écrit alors

$$IC(\lambda) = \left[c_g - \sqrt{\frac{v_g}{N}}\Phi^{-1}\left(1-\frac{\lambda}{2}\right), c_g + \sqrt{\frac{v_g}{N}}\Phi^{-1}\left(1-\frac{\lambda}{2}\right)\right] \tag{3.4}$$

Si v_g correspond bien à un estimateur convergent, sans biais, de $var\left[g\left(X\right)\right]$, le fait de remplacer $var\left[g\left(X\right)\right]$ par v_g dans la formule (3.3) introduit une nouvelle source d'erreur liée à l'aléa de l'échantillonnage.

Notons en outre que l'erreur induite par l'utilisation de la statistique v_g est d'autant plus importante que c_g et v_g ne sont pas des statistiques indépendantes. La corrélation entre ces deux statistiques conduit à sur-estimer, ou sous-estimer, la précision de l'estimateur c_g . Par exemple, dans le cas particulier où $g\left(X\right)$ est une variable de Bernoulli de paramètre λ, Fishman [8] montre que

$$\lim_{N\to\infty} corr\left(c_g, v_g\right) = \begin{cases} 1 & \text{si } 0 \leq \lambda < \frac{1}{2} \\ 0 & \lambda = \frac{1}{2} \\ -1 & \frac{1}{2} < \lambda \leq 1 \end{cases} .$$

En général, la corrélation entre c_g et v_g ne tend donc pas vers 0 lorsque le nombre de simulations utilisées tend vers l'infini.

Les intervalles de confiance que nous présentons dans les divers applications numériques reposent sur l'utilisation de la formule 3.4 page précédente. Les quelques remarques que nous venons de formuler montrent qu'il s'agit d'un outil de contrôle imparfait. Il permet néanmoins d'avoir une première idée de la vitesse de convergence des estimateurs que nous allons construire.

3.2 Variables de contrôle

Cette première méthode repose sur la connaissance d'une fonction h pour laquelle on sait calculer $I_h = \mathbb{E}^{\mathbb{Q}}\left[h\left(X\right)\right]$. On peut alors construire un nouvel estimateur sans biais de I_g exploitant cette information supplémentaire. On considère pour cela l'estimateur c_{ctl} défini de la façon suivante

$$c_{ctl}\left(\beta\right) = c_g - \beta\left(c_h - I_h\right) \tag{3.5}$$

où c_h est l'estimateur de Monte Carlo standard de I_h. La variance de c_{ctl} est

$$var\left(c_{ctl}\right) = var\left(c_g\right) + \beta^2 var\left(c_h\right) - 2\beta cov\left(c_g, c_h\right),$$

si bien que le choix optimal pour β correspond à

$$\beta^* = \frac{cov\left(c_g, c_h\right)}{var\left(c_h\right)},$$

la variance de $c_{ctl}\left(\beta^*\right)$ étant alors égale à $\left(1 - \rho_{g,h}^2\right) var\left(c_g\right)$. Notons que si la variance de $c_{ctl}\left(\beta^*\right)$ est nécessairement inférieure à la variance de c_g, une forte réduction de la variance implique cependant une corrélation très élevée entre la variable $g\left(X\right)$ et la variable de contrôle $h\left(X\right)$, ce qui limite la portée de cette première méthode.

D'un point de vue pratique, β^* étant inconnu (on ne connaît pas *a priori* $cov\left(c_g, c_h\right)$), on construit dans un premier temps un estimateur de ce paramètre à partir d'un nombre relativement réduit de simulations. La valeur obtenue est alors utilisée pour construire un estimateur c_{ctl} de I_g. Notons que l'estimation préliminaire de β^* ne biaise pas l'estimation de I_g par c_{ctl}, que l'on réalise dans un second temps seulement, de façon indépendante.

Dans l'application suivante, l'utilisation d'une variable de contrôle s'avère particulièrement efficace.

Application : le cas d'une option sur un panier d'actions

Le problème que l'on se pose est l'évaluation d'une option d'achat européenne définie à partir de la moyenne arithmétique d'un panier d'actions. Dans un cadre d'analyse Black-Scholes, le prix d'un tel produit ne s'exprime pas de façon analytique. L'estimation du prix de ce type d'option par simulation est très rapide si on utilise comme variable de contrôle le même produit, mais défini cette fois-ci à partir de la moyenne géométrique des prix des actions.

Cadre d'analyse.

Soit $(B_t)_{t\geq 0}$ un mouvement brownien standard sur $\mathbb{R}^d$, $\Sigma = (\sigma_i)_{1\leq i\leq d}$ un vecteur de coordonnées strictement positives et $\Omega = (\rho_{ij})_{1\leq i,j\leq d}$ une matrice de corrélation. On note Γ la matrice associée à Ω résultant de la décomposition de Cholesky[1] de cette dernière.

On définit un processus aléatoire sur $\mathbb{R}^d$, $\left(S_t^1, \ldots, S_t^d\right)_{t\geq 0}$, par l'équation différentielle stochastique suivante :

$$dS_t^i = rdt + \sigma_i dW_t^i, i \in \{1, \ldots, d\}$$

où $W_t = \Gamma B_t, t \geq 0$, de telle sorte que

$$\forall i, j \in \{1, \ldots, N\}, \left\langle W_\cdot^i, W_\cdot^j \right\rangle_t = \rho_{ij} t.$$

Posons $A_t = \frac{1}{d}\sum S_t^i, t \geq 0$, et définissons le processus $(X_t)_{t\geq 0}$ par :

$$X_t = \max\left(A_t - K, 0\right)$$

Le paramètre d'intérêt que l'on souhaite calculer est

$$I_X = \mathbb{E}e^{-rt}X_t.$$

Estimateur standard.

Un premier estimateur sans biais de ce paramètre, noté c_X, peut être obtenu par une méthode de Monte Carlo standard, en simulant N vecteurs $\left(B_t^1, \ldots, B_t^d\right)$ et en calculant les N valeurs $\left(x_t^1, \ldots, x_t^N\right)$ prises par la variable aléatoire X_t. On pose alors :

$$c_X = \frac{e^{-rt}}{N}\sum_{i=1}^{N} x_t^i$$

La variance de cet estimateur est

$$var\left(c_X\right) = \frac{e^{-2rt}}{N} var\left(X_t\right)$$

L'objectif du paragraphe suivant est de construire un nouvel estimateur de I_X dont la variance est inférieure à celle de c_X pour un nombre de simulations N fixé.

Utilisation d'une variable de contrôle.

On définit une variable de contrôle pour l'estimation de I_X en considérant, dans la définition de la fonction X_t à intégrer, non plus la moyenne arithmétique A_t des S_t^i, mais la moyenne géométrique G_t :

$$G_t = \sqrt[d]{\prod_{i=1}^{d} S_t^i}$$

[1] Γ est l'unique matrice triangulaire inférieure telle que $\Gamma\Gamma' = \Omega$.

Si l'on note $Y_t = \max(G_t - K, 0)$, la valeur de $I_Y = \mathbb{E}e^{-rt}Y_t$ est connue de façon analytique. On la calcule en appliquant la formule de Black-Scholes à G_t, G_t pouvant s'écrire sous la forme suivante :

$$G_t = G_0 e^{\left(\tilde{r} - \frac{1}{2}\tilde{\sigma}^2\right)t + \tilde{\sigma}\tilde{B}_t}$$

où $\left(\tilde{B}_t\right)_{t\geq 0}$ est un mouvement brownien standard sur $\mathbb{R}$, et

$$\begin{cases} \tilde{\sigma} = & \frac{1}{d}\sqrt{\Sigma'\Omega\Sigma} \\ \tilde{r} = & r + \frac{1}{2}\tilde{\sigma}^2 - \frac{1}{2d}\sum_{i=1}^{d}\sigma_i^2 \end{cases}$$

Dès lors,

$$I_Y = G_0 e^{(\tilde{r}-r)t}\Phi(d_1) - Ke^{-rt}\Phi(d_2)$$

avec

$$\begin{cases} d_1 = & \frac{\ln\left(\frac{G_0}{K}\right) + \left(\tilde{r} + \frac{1}{2}\tilde{\sigma}^2\right)t}{\tilde{\sigma}\sqrt{t}} \\ d_2 = & d_1 - \tilde{\sigma}\sqrt{t} \end{cases}$$

Applications numériques.

Le jeu de paramètres retenus pour les applications numériques est le suivant : $t = 1$, $r = 0$, $d = 10$, $S0 = \{80, 85, 90, 95, 100, 105, 110, 115, 120, 125\}$, $\Sigma = \{0.15,\ 0.16,\ 0.17,\ 0.18,\ 0.19,\ 0.20,\ 0.21,\ 0.22,\ 0.23,\ 0.24\}$, $\rho_{ii} = 1$, $\rho_{ij} = 0.5$ si $i \neq j$. On fait varier le prix d'exercice K afin d'étudier l'efficacité de la méthode selon que l'on se situe à la monnaie, dans la monnaie ou hors la monnaie :

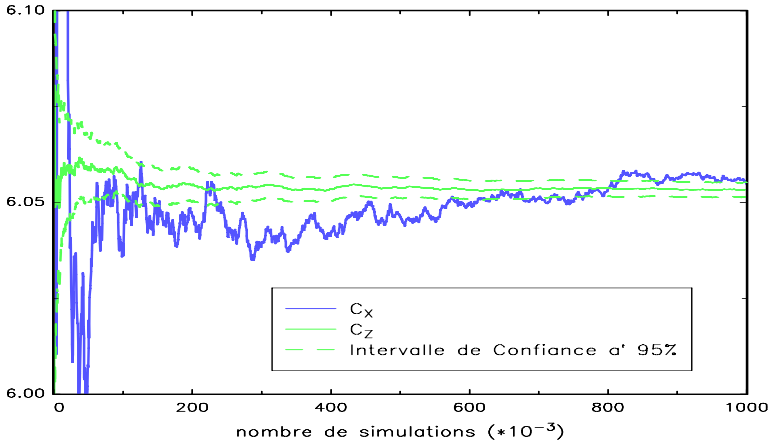
$$K \in \left\{ \left(\frac{1}{10}\sum_{i=1}^{10} S0_i = 102.5 \right), 50, 150 \right\}.$$

Les graphiques 3.1 page suivante, 3.2 page suivante et 3.3 page suivante illustrent le comportement des estimateurs contrôlés pour chacune de ces trois valeurs.

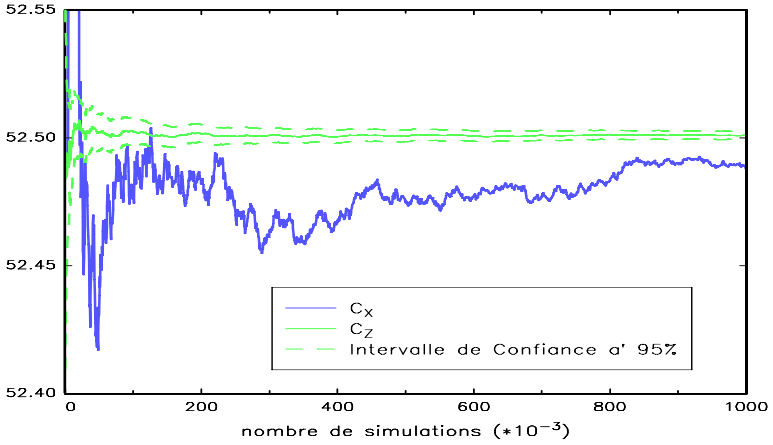
$K = 102.5$ **:** Le coefficient de corrélation entre les variables X_t et Y_t est estimé à 99.5%. La valeur induite pour le coefficient β est de 1.12.
La variance de l'estimateur contrôlé c_Z est ici 100 fois plus petite que la variance de l'estimateur standard c_X.

$K = 50$ **:** Le coefficient de corrélation entre les variables X_t et Y_t est estimé à 99.86%. La valeur induite pour le coefficient β est de 1.042.
La variance de l'estimateur contrôlé c_Z est 360 fois plus petite que la variance de l'estimateur standard c_X.

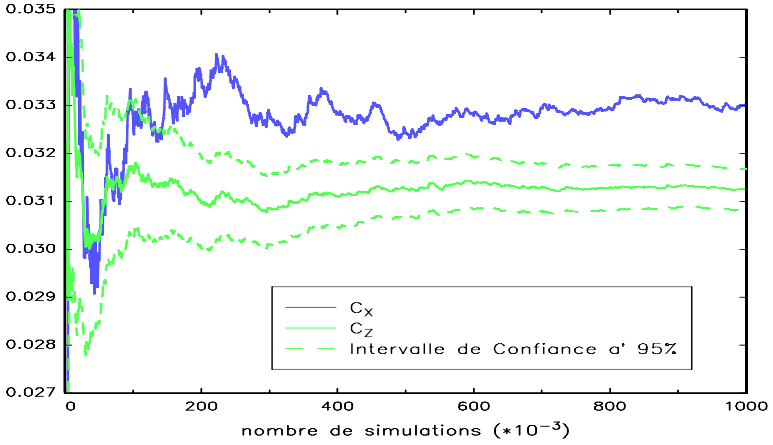
$K = 150$ **:** Le coefficient de corrélation entre les variables X_t et Y_t est ici beaucoup plus faible que dans les cas précédents : il est estimé à 95%. La valeur induite pour le coefficient β est de 1.4.
La variance de l'estimateur contrôlé c_Z est 10 fois plus petite que la variance de l'estimateur standard c_X. La méthode demeure donc intéressante.



Graphique 3.1. Accélération par variable de contrôle, $K = 102.5$, $\beta = 1.12$



Graphique 3.2. Accélération par variable de contrôle, $K = 50$, $\beta = 1.042$



Graphique 3.3. Accélération par variable de contrôle, $K = 150$, $\beta = 1.4$

3.3 Échantillonnage d'importance

3.3.1 Méthode générale

L'échantillonnage d'importance constitue une deuxième méthode permettant de construire des estimateurs du paramètre d'intérêt I_g (équation 3.1 page 29), dont la variance, pour un nombre de simulations donné, est inférieure à la variance de l'estimateur de Monte Carlo standard. L'idée de l'échantillonnage d'importance est d'utiliser non plus la loi μ pour la simulation des variables aléatoires, mais une loi instrumentale qui tient compte de la répartition, au sein de l'espace $\mathcal{S}$, du produit $g(x)\,\mu(dx)$. Les simulations ainsi obtenues sont alors pondérées pour supprimer le biais induit par le changement de loi lors de l'étape de simulation.

Supposons que l'espace $\mathcal{S}$ puisse être identifié à $\mathbb{R}^d, d \geq 1$, et que la mesure de probabilité μ admette une densité f par rapport à la mesure de Lebesgue sur $\mathbb{R}^d$:

$$\mu(dx) = f(x)\,dx.$$

De plus, en limitant le champ de l'étude aux fonctions g positives, on peut réécrire I_g sous la forme

$$I_g = \int_{\mathbb{R}^d} g(x)\,\mathbf{1}_D(x)\,f(x)\,dx \tag{3.6}$$

où D représente le domaine au sein duquel la fonction g prend des valeurs strictement positives.

Considérons à présent une fonction de densité sur $\mathbb{R}^d$, notée h , telle que

$$\forall x \in \mathbb{R}^d, g(x) > 0 \Longrightarrow h(x) > 0.$$

Le relation (3.6) peut se réécrire

$$I_g = \int_{\mathbb{R}^d} g(x)\,\frac{f(x)}{h(x)}\mathbf{1}_D(x)\,h(x)\,dx. \tag{3.7}$$

Il apparaît alors clairement que l'on peut construire un nouvel estimateur c_{is} du paramètre I_g en simulant N variables aléatoires indépendantes $y_1, \ldots, y_N$ suivant la loi h et en posant

$$c_{is} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(y_i)}{h(y_i)}g(y_i) \tag{3.8}$$

La variance de cet estimateur vaut

$$var(c_{is}) = \int_D \left(g(x)\,\frac{f(x)}{h(x)} - I_g\right)^2 h(x)\,dx.$$

Ainsi la densité h^* qui minimise la variance de l'estimateur pondéré s'écrit :

$$h^*(x) = \frac{g(x)\,f(x)}{I_g},$$

la variance de l'estimateur c_{is}^* étant alors nulle. Notons que ce résultat est sans grand intérêt pratique puisque la construction de h^* fait intervenir le paramètre d'intérêt I_g ! Il montre néanmoins que la densité h à utiliser pour la construction de c_{is} doit être choisie de telle sorte que $\frac{fg}{h}$ soit quasi constant et de variance finie. Le changement de loi conduit donc à un tirage fréquent des valeurs de x pour lesquelles le produit gf prend des valeurs élevées, les valeurs de x telles que $f(x)\,g(x)$ est faible étant plus rarement sélectionnées.

Application

Nous reprenons ici l'exemple introduit à la sous-section 3.2 page 31. Le paramètre d'intérêt I_X peut s'écrire

$$I_X = \int_{\mathcal{D}} X_t(z)\,\phi(z)\,dz$$

où ϕ représente la densité de la loi normale centrée réduite en dimension d, et $\mathcal{D}$ le domaine de $\mathbb{R}^d$ sur lequel X_t est strictement positif. Pour toute fonction de densité h telle que

$$\forall z \in \mathcal{D},\ \text{si } \phi(z) > 0,\ \text{alors } h(z) > 0,$$

on peut réécrire I_X sous la forme

$$I_X = \int_{\mathcal{D}} X_t(z)\,\frac{\phi(z)}{h(z)}h(z)\,dz.$$

Ainsi, en considérant N réalisations indépendantes $(z_1, \ldots z_N)$ d'une variable aléatoire distribuée sur $\mathbb{R}^d$ suivant la loi h, on construit un nouvel estimateur sans biais de I_X en posant

$$c_h = \frac{1}{N}\sum_{i=1}^{N}\frac{\phi(z_i)}{h(z_i)}X_t(z_i).$$

Cet estimateur a pour variance

$$\begin{aligned} var(c_h) &= \frac{1}{N}\int_{\mathcal{D}}\left(\frac{\phi(z)}{h(z)}X_t(z) - I_X\right)^2 h(z)\,dz. \\ &= \frac{1}{N}\int_{\mathcal{D}}\frac{\phi(z)}{h(z)}X_t^2(z)\,\phi(z)\,dz - \frac{1}{N}I_X^2 \end{aligned}$$

Il convient donc de choisir h de telle sorte que $V(h) = \mathbb{E}_\phi \frac{\phi(Z)}{h(Z)}X_t^2(Z)$ soit le plus petit possible.

Choix de la fonction d'importance.

L'obtention de la fonction h optimale étant impossible, on restreint le domaine des fonctions admissibles aux densités des lois normales $\mathcal{N}(\mu, I_d)$, $\mu \in \mathbb{R}^d$. On souhaite donc déterminer le vecteur μ qui rend $V(h) = \mathbb{E}\frac{\phi(Z)}{h(Z)}X_t^2(Z)$ le plus petit possible, avec

$$\frac{\phi(Z)}{h(Z)} = \exp\left(\frac{1}{2}\mu'\mu - \mu' Z\right).$$

On ne cherchera pas ici à résoudre ce programme. On utilisera plutôt le raisonnement heuristique suivant :

Pour qu'un échantillonnage d'importance soit efficace, il faut que la loi instrumentale permette de nombreux tirages dans les regions de $\mathcal{D}$ où $\phi(z)\,X_t(z)$ prend des valeurs élevées, les tirages dans les régions où $\phi(z)\,X_t(z)$ prend des petites valeurs étant alors plus rares. La pondération par $\frac{\phi(z)}{h(z)}$ permet de "lisser" le résultat : les termes tels que $\phi(z)\,X_t(z)$ est petit sont associés à une petite valeur de $h(z)$, et inversement, les termes tels que $\phi(z)\,X_t(z)$ est élevé sont associés à une valeur élevée de $h(z)$.

On choisira donc le vecteur μ comme solution de

$$\max_{z\in\mathcal{D}} \ln X_t(z) - \frac{1}{2}z'z$$

Glasserman [13] montre "qu'asymptotiquement", il est équivalent de résoudre ce dernier programme ou de minimiser $V(h)$.

L'estimateur c_μ que l'on considère dans les applications numériques suivantes s'écrit :

$$c_\mu = \frac{1}{N} \sum_{i=1}^{N} e^{\frac{1}{2}\mu' \mu - \mu' z_i} X_t(z_i)$$

où $(z_1, \ldots, z_N)$ sont N réalisations indépendantes d'une loi normale $\mathcal{N}(\mu, I_d)$.

Applications.

On considère à nouveau les trois valeurs de K fixées à la sous-section 3.2 page 31. De même que pour l'estimateur contrôlé, l'estimateur pondéré conduit à une réduction de variance plus ou moins importante selon la valeur de K considérée. Les graphiques 3.4 page suivante, 3.5 page suivante et 3.6 page suivante illustrent le comportement des estimateurs pondérés en fonction du nombre de simulations.

$K = 102.5$ **:** La variance de l'estimateur pondéré c_μ est 9 fois plus petite que celle de l'estimateur standard c_X.

$K = 50$ **:** La variance de l'estimateur pondéré c_μ est 90 fois plus petite que celle de l'estimateur standard c_X.

$K = 150$ **:** La variance de l'estimateur pondéré c_μ est 400 fois plus petite que celle de l'estimateur standard c_X. On constate qu'ici, l'échantillonnage d'importance est particulièrement efficace. Ceci est dû au fait que le domaine $\mathcal{D}$ sur lequel $X_t > 0$ correspond à un événement rare dans la mesure où $A_0 = 102.5$, ce qui est nettement inférieur à $K = 150$. L'utilisation d'une loi normale centrée sur 0 pour la simulation de X_t ne permet que très rarement d'atteindre le domaine $\mathcal{D}$. Le décentrage par l'intermédiaire du vecteur μ est particulièrement efficace car il accroît de façon conséquente le nombre de tirages au sein du domaine $\mathcal{D}$.

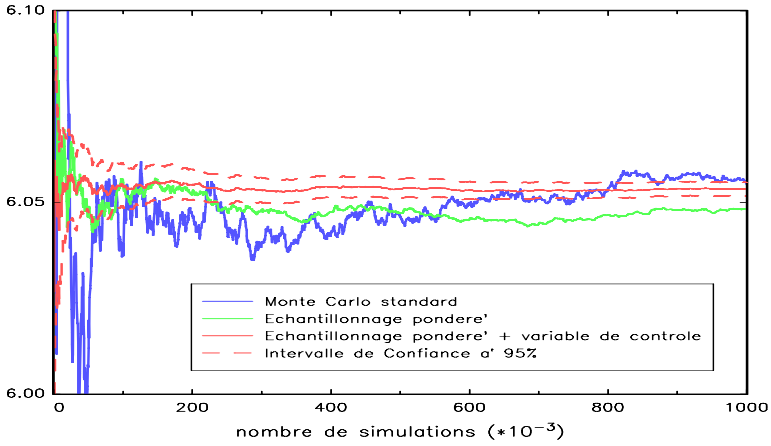
Échantillonnage d'importance et variable de contrôle.

Dans ce dernier paragraphe, on construit un estimateur à partir des deux méthodes que l'on vient de présenter : on applique la technique d'échantillonnage d'importance de façon simultanée à X_t et à la variable de contrôle Y_t. Le graphiques 3.4 page suivante, 3.5 page suivante 3.6 page suivante rendent compte de la vitesse de convergence d'un tel estimateur toujours pour les trois mêmes valeurs de K.

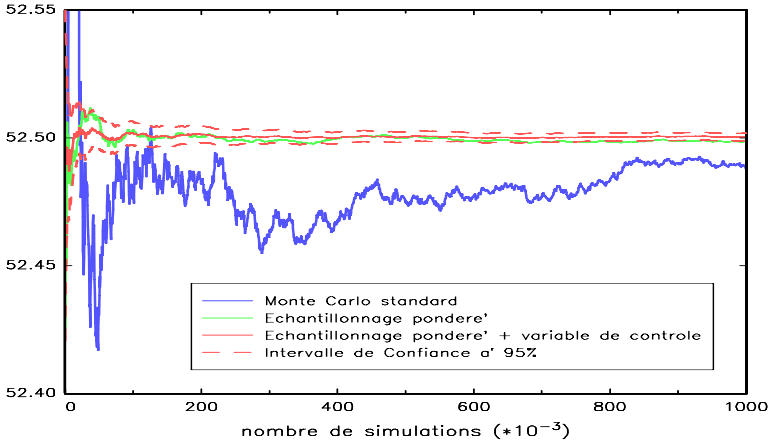
$K = 102.5$ **:** Le coefficient de corrélation entre les deux variables pondérées X_t et Y_t est estimé à 96%, ce qui induit une valeur de 1.077 pour β. La variance de l'estimateur ainsi construit est 118 fois plus petite que la variance de l'estimateur standard. Notons que dans ce premier cas, l'échantillonnage d'importance ne permet pas d'améliorer de façon très sensible les performances de l'estimateur contrôlé.

$K = 50$ **:** Le coefficient de corrélation entre les deux variables pondérées est estimé à 89.05%, ce qui induit une valeur de 0.875 pour β. La variance de l'estimateur est réduite dans un rapport de 435. Le cumul des deux méthodes s'avère donc ici plutôt efficace. L'estimateur pondéré ayant déjà permis à lui seul une réduction de variance de 90, l'ajout de la variable de contrôle permet de multiplier cette valeur par 5 environ.

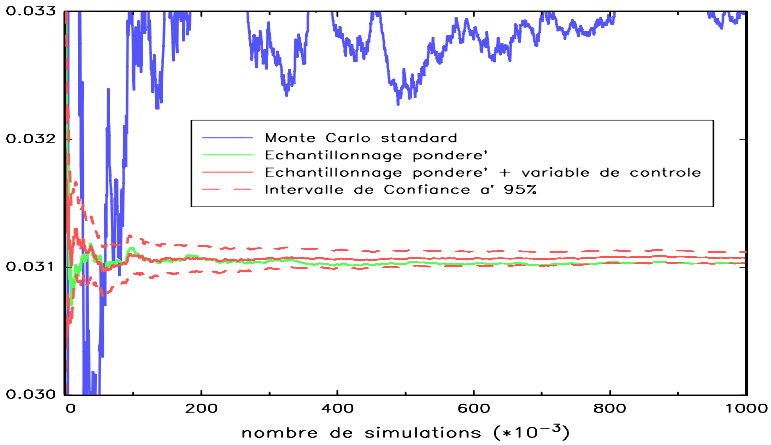
$K = 150$ **:** Le coefficient de corrélation entre les deux variables pondérées est estimé à 75.6%. Le coefficient β vaut 1.4. La variable de contrôle ne permet une réduction de variance qui n'est que de l'ordre de 2.3 par rapport à l'estimateur pondéré seul. Néanmoins, ce dernier induisant déjà une réduction de variance de l'ordre de 400, l'estimateur construit à partir des deux méthodes possède une variance 900 fois plus petite que celle de l'estimateur standard.



Graphique 3.4. Échantillonnage d'importance et cumul des méthodes, $K = 102.5$



Graphique 3.5. Échantillonnage d'importance et cumul des méthodes, $K = 50$



Graphique 3.6. Échantillonnage d'importance et cumul des méthodes, $K = 150$

3.3.2 Le cas des processus de diffusion

Le choix du changement de probabilité pour la simulation des variables aléatoires constitue la difficulté principale des méthodes d'échantillonnage d'importance. La famille de densités possibles, au sein de laquelle on recherchera la loi instrumentale, dépend bien entendu du cadre d'analyse dans lequel on se place.

Dans le cas particulier des modèles financiers, les variables aléatoires utilisées correspondent la plupart du temps à des processus de diffusion. Ainsi, I_g s'écrit

$$I_g = I_g(t,x) = \mathbb{E}^{\mathbb{Q}}\left[g\left(X_s, t \leq s \leq T\right) \mid X_t = x\right],$$

$X = (X_t)_{0\leq t\leq T}$ étant un processus à valeur dans $\mathbb{R}^d$, solution d'une équation différentielle stochastique de la forme

$$dX_t = b(t, X_t)\, dt + \sigma(t, X_t)\, dW_t, \tag{3.9}$$

où $(W_t)_{t\geq 0}$ est un mouvement brownien standard de dimension d défini sur l'espace $(\Omega, \mathcal{F}, \mathbb{Q})$. Le théorème de Girsanov que nous rappelons ici permet d'introduire un nouveau formalisme afin d'orienter le choix de la loi instrumentale.

Théorème 3.1 – *Girsanov* –

Soit $T > 0$, $\left(\Omega, (\mathcal{F}_t)_{0\leq t\leq T}, \mathcal{F}_T, \mathbb{Q}\right)$ un espace filtré et (W_t) un $\mathcal{F}_t$ – mouvement brownien d - dimensionnel issu de 0.

Soit $(\phi_t)_{0\leq t\leq T}$ un processus à valeurs dans $\mathbb{R}^d$, mesurable, adapté. Si le processus $M^{(\phi)}$ défini par

$$M_t^{(\phi)} = \exp\left[-\int_0^t \phi_s dW_s - \frac{1}{2}\int_0^t |\phi_s|^2\, ds\right],$$

est une martingale sous $\mathbb{Q}$ alors on définit une nouvelle probabilité $\mathbb{Q}^{(\phi)}$ sur $(\Omega, \mathcal{F}_T)$ en posant[2] :

$$\frac{d\mathbb{Q}^{(\phi)}}{d\mathbb{Q}} = M_T^{(\phi)},$$

et le processus $B_t = W_t + \int_0^t \phi_s ds, t \in [0,T]$ est un $\mathcal{F}_t$-mouvement brownien pour la probabilité $\mathbb{Q}^{(\phi)}$.

Compte tenu de ce théorème, il apparaît que le paramètre d'intérêt $I_g(t,x)$ peut s'écrire :

$$I_g(t,x) = \mathbb{E}^{\mathbb{Q}^{(\phi)}}\left[g\left(X_s, t \leq s \leq T\right) L_{t,T}^{(\phi)} \mid X_t = x\right], \tag{3.10}$$

où

$$\begin{aligned} L_{t,T}^{(\phi)} &= \tfrac{M_t^{(\phi)}}{M_T^{(\phi)}} \\ &= \exp\left[\int_t^T \phi_s dB_s - \tfrac{1}{2}\int_t^T |\phi_s|^2\, ds\right] \end{aligned}$$

Un estimateur $c_{is}^{(\phi)}(t,x)$ de $I_g(t,x)$ peut donc être construit en simulant N trajectoires indépendantes suivant la loi du processus $(X_s)_{t\leq s\leq T}$ sous la probabilité $\mathbb{Q}^{(\phi)}$, et en posant

$$c_{is}^{(\phi)}(t,x) = \frac{1}{N}\sum_{i=1}^{N} g\left(x_s(i), t \leq s \leq T\right) L_{t,T}^{(\phi)}(i). \tag{3.11}$$

[2] Une condition suffisante portant sur le processus ϕ correspond au critère de Novikov : si $\mathbb{E}^{\mathbb{Q}}\left[\exp\left(\int_0^T |\phi_s|^2\, ds\right)\right] < \infty$ alors $\mathbb{E}^{\mathbb{Q}}\left[M_T^{(\phi)}\right] = 1$ et $M^{(\phi)}$ est une $\mathbb{Q}$ – martingale.

La simulation de trajectoires suivant la loi du processus $(X_s)_{t\leq s\leq T}$ sous la probabilité $\mathbb{Q}^{(\phi)}$ peut être réalisée en remarquant que $(X_s)_{t\leq s\leq T}$, solution de l'équation différentielle stochastique (3.9) sous $\mathbb{Q}$, est la solution, sous $\mathbb{Q}^{(\phi)}$, de l'équation différentielle stochastique suivante :

$$dX_s = [b(s, X_s) - \sigma(s, X_s)\phi_s]\, dt + \sigma(s, X_s)\, dB_s. \tag{3.12}$$

Le changement de probabilité pour la construction d'un nouvel estimateur $c_{is}^{(\phi)}(t,x)$ de $I_g(t,x)$ conduit donc à la modification du processus de dérive intervenant dans la caractérisation du processus (X_t). L'idée est ici de modifier l'allure des trajectoires par le choix d'un processus ϕ judicieux, afin de prendre en compte les spécificités de la fonction g à intégrer.

Processus optimal ϕ^*.

Si l'on suppose que la fonction g ne dépend que de la valeur terminale X_T du processus sous-jacent, il est alors possible d'exhiber un changement de dérive optimal, conduisant à une variance nulle. Le paramètre d'intérêt $I_g(t,x)$ se réécrit, sous la probabilité $\mathbb{Q}^{(\phi)}$ induite par le processus ϕ :

$$I_g(t,x) = \mathbb{E}^{\mathbb{Q}^{(\phi)}}\left[g(X_T)\, L_{t,T}^{(\phi)} \mid X_t = x\right].$$

On peut alors montrer, en appliquant le calcul d'Itô, que la variable $g(X_T)\, L_{t,T}^{(\phi)}$ s'écrit :

$$g(X_T)\, L_{t,T}^{(\phi)} = I_g(t,x) + \int_t^T L_{t,s}^{(\phi)} \left[\sigma(X_s)' \nabla_x I_g(s, X_s) + \phi_s I_g(s, X_s)\right] dB_s$$

où $\nabla_x I_g$ représente le gradient de la fonction I_g par rapport aux coordonnées de X. Il apparaît donc que la variance de $g(X_T)\, L_{t,T}^{(\phi)}$ sera nulle dès que

$$\phi_s = \phi_s^* = -\frac{1}{I_g(s, X_s)} \sigma(X_s)' \nabla_x I_g(s, X_s). \tag{3.13}$$

La fonction I_g étant inconnue, il est impossible de mettre en œuvre un changement de dérive utilisant ϕ^*. Néanmoins, l'utilisation d'un processus ϕ construit à partir de la formule (3.13), en remplaçant I_g par une fonction qui l'approche, peut permettre une réduction significative de la variance. Notons de plus que si la formule (3.13) a été établie pour des fonctions g ne dépendant que de X_T, on peut supposer que son utilisation, dans le cadre plus général de fonctions dépendant de tout ou partie de la trajectoire, induira également une réduction de la variance.

Application : le modèle de Black et Scholes comme approximation des processus à volatilité stochastique

On remplace la fonction $I_g(t,x)$ dans la formule (3.13) par la fonction $I_g(t,x)$ que l'on obtient lorsque l'on spécifie un nouveau modèle sous-jacent pour le processus (X_s) "approximant" le modèle initial, et dans lequel $I_g(t,x)$ s'écrit sous la forme d'une formule fermée. Le modèle de Black et Scholes constitue un candidat naturel pour appliquer ce type de méthode.

Considérons le cas du pricing d'un call européen standard dans le cadre du modèle de Heston [14]. Il s'agit d'un modèle à volatilité stochastique : le processus sous-jacent $(X_t) = (S_t, V_t)$ est à valeurs dans $\mathbb{R}^2$, solution de l'équation différentielle stochastique

$$\begin{aligned} \frac{dS_t}{S_t} &= r dt + \sqrt{V_t} dW_{1,t} \\ dV_t &= \kappa(\theta - V_t)\, dt + \gamma\sqrt{V_t} dW_{2,t} \end{aligned}$$

où $(W_{1,t})$ et $(W_{2,t})$ sont deux mouvements browniens standards unidimensionnels tels que

$$\langle dW_{1,t}, dW_{2,t}\rangle = \rho dt.$$

Notons T la maturité du call européen. La méthode de Monte Carlo standard consiste à simuler N trajectoires du processus (X_t) entre 0 et T en utilisant un schéma d'Euler de la forme :

$$\begin{aligned} S_{t_{i+1}} &= S_{t_i}\left(1 + r\Delta t + \sqrt{V_{t_i}}\sqrt{\Delta t}Z^1_{i+1}\right) \\ V_{t_{i+1}} &= V_{t_i} + \kappa\left(\theta - V_{t_i}\right)\Delta t + \gamma\sqrt{V_{t_i}}\sqrt{\Delta t}\left(\rho Z^1_{i+1} + \sqrt{1-\rho^2}Z^2_{i+1}\right) \end{aligned}$$

où $\Delta t = t_{i+1} - t_i$ représente le pas de discrétisation et $\left(Z^1_i, Z^2_i\right)$, $i \in \{1, ..., m\}$, sont $m = \frac{T}{\Delta t}$ lois iid $\mathcal{N}(0, I_2)$.

Remarque 3.1 *Il est possible de rendre l'écart entre le processus $(X_s)_{t\leq s\leq T}$ et le processus discrétisé aussi petit qu'on le souhaite. Il suffit pour cela de choisir un pas de discrétisation Δt suffisamment fin (cf [17,* Kloeden *et* Platen*]).*

Si l'on note S^1_T,...,S^N_T les N simulations du prix du sous-jacent ainsi obtenues, l'estimateur de Monte Carlo standard c^0 s'écrit

$$c^0 = \frac{1}{N}\sum_{i=1}^{N} e^{-rT}(S^i_T - K)^+$$

K étant le prix d'exercice de l'option considérée.

L'accélération de la convergence est réalisée au moyen d'un changement de dérive ϕ calculée à partir de la formule (3.13) en approchant le paramètre d'intérêt I_g et son gradient à partir des formules classiques du prix et des grecs d'un call européen dans le modèle de Black-Scholes[3].

Le processus $(X_t) = (S_t, V_t)$ est alors solution, sous la nouvelle probabilité, de l'équation différentielle stochastique

$$dX_t = \left[b(t, X_t) - \Sigma(t, X_t)\phi_t\right]dt + \Sigma(t, X_t)dB_t$$

avec

$$\Sigma(t, X_t) = \begin{pmatrix} X_{1,t}\sqrt{X_{2,t}} & 0 \\ \rho\gamma\sqrt{X_{2,t}} & \sqrt{1-\rho^2}\gamma\sqrt{X_{2,t}} \end{pmatrix} \text{ et } b(t, X_t) = \begin{pmatrix} rX_{1,t} \\ \kappa\left(\theta - X_{2,t}\right) \end{pmatrix}.$$

La simulation du processus (X_t) sous la nouvelle probabilité s'effectue à nouveau en utilisant un schéma d'Euler. De même, le calcul du poids $L^{(\phi)}_T$ à associer à chacune des simulations est réalisé en approchant $\int_0^T \phi_s dBs$ et $\int_0^T |\phi_s|^2 ds$.

Notons à nouveau $t_0 = 0$, t_1,...., $t_m = T$ les différents instants intervenant dans le schéma de discrétisation. $L^{(\phi)}_T$ est obtenu en remarquant que

$$\begin{aligned} L^{(\phi)}_{t_{i+1}} &= exp\left[\int_{t_i}^{t_{i+1}} \phi_s dBs - \frac{1}{2}\int_{t_i}^{t_{i+1}} |\phi_s|^2 ds\right] L^{(\phi)}_{t_i} \\ L^{(\phi)}_0 &= 1. \end{aligned}$$

[3]Dans le modèle Black-Scholes, le prix, le delta et le vega d'un call européen de maturité T et de strike K, valent à la date t, en fonction du prix s et de la volatilité σ :

$$\begin{aligned} &CALL\left(t, s, K, T, \sigma\right) = s\Phi\left(d_1\right) - Ke^{-r(T-t)}\Phi\left(d_1 - \sigma\sqrt{T-t}\right) \\ &\Delta_{CALL}\left(t, s, T, K, \sigma\right) = \Phi\left(d_1\right) \\ &\upsilon_{CALL}\left(t, s, T, K, \sigma\right) = s\sqrt{T-t}\Phi'\left(d_1\right) \end{aligned}$$

avec $d_1 = \frac{\ln\left(\frac{s}{Ke^{-r(T-t)}}\right) + \frac{1}{2}\sigma^2(T-t)}{\sigma\sqrt{T-t}}$.

On peut donc approcher $L_T^{(\phi)}$ en considérant le processus discrétisé suivant :

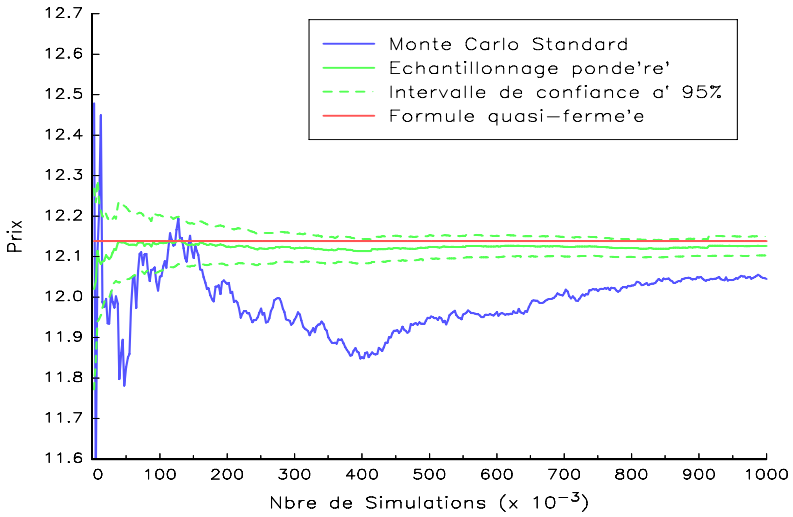
$$\begin{aligned} L_{t_{i+1}}^{(\phi)} &= exp\left[\sqrt{\Delta t}\,\langle \phi_{t_i}, Z_{i+1}\rangle - \frac{\Delta t}{2}|\phi_{t_i}|^2\right] L_{t_i}^{(\phi)} \\ L_0^{(\phi)} &= 1. \end{aligned}$$

où $\left(Z_i^1, Z_i^2\right)$, $i \in \{1, ..., m\}$, sont les m lois iid $\mathcal{N}(0, I_2)$ intervenant dans la simulation de X_{t_1},..., X_{t_m}.

Si l'on note $S_1^{(\phi)}(T)$,..., $S_N^{(\phi)}(T)$ les N prix du sous-jacent ainsi simulés, et $L_1^{(\phi)}(T)$,..., $L_N^{(\phi)}(T)$ les poids associés, l'estimateur avec changement de dérive s'écrit

$$c^{(\phi)} = \frac{1}{N}\sum_{i=1}^{N} e^{-rT}(S_i^{(\phi)}(T) - K)^+ L_i^{(\phi)}(T).$$

Le graphique 3.7 illustre les performances de cette méthode dans le cas d'un call à la monnaie.



Graphique 3.7. Prix d'un call européen dans le modèle de Heston : $S_0 = 100$, $V_0 = 0.3^2$, $K = 100$, $T = 1$.

Les paramètres du modèle sont les suivants : $r = 5\%$, $\kappa = 2$, $\theta = 0.2^2$, $\gamma = 0.2$ et $\rho = 0.5$. Le pas de discrétisation est $\Delta t = \frac{1}{365}$.

3.4 Stratification

Supposons maintenant que l'on puisse découper l'espace $\mathcal{S}$ des valeurs de X en k ensembles mesurables Str_i, appelés strates, pour lesquels on sait calculer

$$\mu(Str_i) = \mathbb{Q}(X \in Str_i) = \mathbb{Q}(Str(X) = Str_i)$$

où $Str(X)$ représente la variable aléatoire indicatrice de la strate dans laquelle se situe X. Si de plus, on est capable de générer les variables aléatoires conditionnelles $X \mid (Str(X) = Str_i)$, de lois

$$\mu_i(dx) = \frac{\mathbf{1}_{Str_i}(x)}{\mu(Str_i)}\mu(dx), i \in \{1, \ldots, k\},$$

il peut être intéressant de récrire le paramètre d'intérêt $I_g = \mathbb{E}^{\mathbb{Q}}[g(X)]$ sous la forme équivalente suivante :

$$I_g = \sum_{i=1}^{k} \mu(Str_i)\, \mathbb{E}^{\mathbb{Q}}[g(X) \mid X \in Str_i]$$

où

$$\mathbb{E}^{\mathbb{Q}}[g(X) \mid Str_i] = \frac{1}{\mu(Str_i)} \int_{Str_i} g(x)\, \mu(dx).$$

La méthode de Monte Carlo par stratification consiste alors à simuler, au sein de chacune des strates Str_i considérée, un échantillon de N_i réalisations indépendantes $\left(x_1^i, \ldots, x_{N_i}^i\right)$ issues de variables aléatoires distribuées suivant μ_i. On construit alors un estimateur sans biais de I_g en posant

$$c_{str} = \sum_{i=1}^{k} \frac{\mu(Str_i)}{N_i} \sum_{j=1}^{N_i} g\left(x_j^i\right) \tag{3.14}$$

Posons $N = \sum_{i=1}^{k} N_i$ et supposons que $N_i = \mu(Str_i)\, N$. Dans ce cas, la variance de l'estimateur c_{str} s'écrit :

$$var(c_{str}) = \frac{1}{N} \mathbb{E}^{\mathbb{Q}}[var(g(X) \mid Str(X))].$$

La formule de décomposition de la variance permet d'écrire

$$var(g(X)) = \mathbb{E}^{\mathbb{Q}}[var(g(X) \mid Str(X))] + var\left(\mathbb{E}^{\mathbb{Q}}[g(X) \mid Str(X)]\right),$$

si bien que

$$var(c_{str}) = var(c_g) - \frac{1}{N} var\left(\mathbb{E}^{\mathbb{Q}}[g(X) \mid Str(X)]\right).$$

La stratification permet donc de réduire la variance, par rapport à une méthode de Monte Carlo standard, du terme $\frac{1}{N} var\left(\mathbb{E}^{\mathbb{Q}}[g(X) \mid Str(X)]\right)$. Notons que cette méthode sera d'autant plus efficace que la variance de la variable $g(X)$ au sein de chacune des strates sera faible, et que la variance inter-strate sera élevée.

Le cas d'un vecteur gaussien.

On s'intéresse ici au cas particulier d'un vecteur gaussien. Considérons une variable aléatoire X à valeur dans $\mathcal{S} = \mathbb{R}^d$, de loi $\mathcal{N}(0, I_d)$ et un vecteur unitaire $s \in \mathbb{R}^d$. Le vecteur aléatoire X peut se récrire sous la forme

$$X = (s'X)\, s + Y$$

où $Y = X - (s'X)\, s$ est un vecteur gaussien centrée de matrice de variance-covariance $I_d - P_s$, $P_s = ss'$ étant la matrice de la projection orthogonale sur le vecteur s. De plus, Y est indépendant de $s'X \sim \mathcal{N}(0,1)$[4].

On construit alors k strates $Str_i, i \in \{1, \ldots, k\}$, suivant la valeur prise par la variable $s'X$. Plus précisément, on définit Str_i par

$$\forall i \in \{1, \ldots, k\}, X \in Str_i \Longleftrightarrow \frac{i-1}{k} \leq \Phi(s'X) < \frac{i}{k}$$

[4]Rappelons que si

$$(X_1, X_2) \sim \mathcal{N}\left[\begin{pmatrix} m_1 \\ m_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}\right],$$

alors

$$X_2 \,|X_1 = x_1 \sim \mathcal{N}\left[m_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1 - m_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right].$$

Le résultat présenté s'obtient en considérant $X_2 = X$ et $X_1 = s'X$.

où Φ représente la fonction de répartition de la loi normale centrée réduite. On tire alors le vecteur aléatoire $X \mid X \in Str_i$ à partir de la décomposition :

$$X \mid X \in Str_i \overset{loi}{=} \Phi^{-1}(u_i)\, s + Y$$

où u_i suit une loi uniforme sur $\left[\frac{i-1}{k}; \frac{i}{k}\right]$ indépendante de Y.

Le problème qui se pose alors est de déterminer une direction s permettant de réduire significativement la variance. Pour plus de détail sur la façon de déterminer s, le lecteur pourra se référer à [13, Glasserman et al., 1998].

3.5 Variables antithétiques

Jusqu'à présent, nous avons présenté les méthodes de Monte Carlo comme des méthodes nécessitant la production d'échantillons de variables aléatoires indépendantes, identiquement distribuées. Néanmoins, il peut s'avérer préférable de construire des échantillons de variables corrélées dans la mesure où cela peut conduire à une réduction de la variance des estimateurs correspondants.

Plus précisément, considérons un premier échantillon de taille $2N$, $(x_1, \ldots, x_{2N})$, de variables aléatoires indépendantes distribuées suivant la loi μ. Cet échantillon permet d'estimer le paramètre d'intérêt $I_g = \int_{\mathcal{S}} g(x)\, \mu(dx)$ à partir de la méthode de Monte Carlo standard, en utilisant la statistique c_g. Si l'on dispose par ailleurs d'un second échantillon de taille N, $(y_1, \ldots y_N)$, de variables aléatoires iid suivant la loi μ, on peut construire un nouvel estimateur c_{ant} de I_g en posant :

$$c_{ant} = \frac{1}{2N} \sum_{i=1}^{N} g(x_i) + g(y_i)\,. \tag{3.15}$$

Si les variables $g(X_i)$ et $g(Y_i)$ sont corrélées négativement, cet estimateur est plus efficace que l'estimateur de Monte Carlo standard fondé sur un échantillon iid de taille $2N$.

Rubinstein [22] propose la méthode suivante pour la simulation des échantillons $(x_i)_{1 \leq i \leq N}$ et $(y_i)_{1 \leq i \leq N}$ dans le cas où $g \circ \mu^-$ est monotone (μ^- représente l'inverse de la fonction de répartition de la loi μ) : x_i est généré à partir de la réalisation d'une loi uniforme u_i en posant

$$x_i = \mu^-(u_i)\,,$$

y_i étant généré de la même façon à partir de $1 - u_i$. Cette méthode est relativement restrictive dans la mesure où d'une part, il faut que la condition de monotonie portant sur $g \circ \mu^-$ soit vérifiée, et d'autre part, il faut disposer de μ^- de façon analytique, ce qui est rarement le cas.

Une méthode plus aisément implémentable a été proposée par Geweke [12] dans le cas où la loi μ est symétrique autour d'une valeur m. Cette méthode consiste à simuler un échantillon de N réalisations $(x_1, \ldots, x_N)$ de façon indépendante suivant la loi μ et à poser $y_i = 2m - x_i, i \in \{1, \ldots, N\}$. Une telle méthode est facilement applicable en finance dans la mesure où les variables aléatoires simulées sont la plupart du temps construites à partir de lois normales.

4

Utilisation des copules pour la simulation d'un vecteur aléatoire

Dans ce chapitre, nous abordons le problème de la simulation du vecteur aléatoire $\mathbf{X} = (X_1, \ldots, X_n)$ à partir de la représentation copule de sa distribution $\mathbf{F}$:

$$\mathbf{F}(x_1, \ldots, x_n) = \mathbf{C}(\mathbf{F}_1(x_1), \ldots, \mathbf{F}_n(x_n))$$

Ce problème revient à simuler le vecteur aléatoire $\mathbf{U} = (U_1, \ldots, U_n)$ dont la distribution est la copule $\mathbf{C}$ et à utiliser la transformation $\mathbf{X} = \left(\mathbf{F}_1^{-1}(U_1), \ldots, \mathbf{F}_n^{-1}(U_n)\right)$. La difficulté majeure est donc de simuler la copule $\mathbf{C}$. Nous présentons trois méthodes pour obtenir des nombres aléatoires de $\mathbf{C}$. Parfois, les marges $\mathbf{F}_1, \ldots, \mathbf{F}_n$ ne sont pas connues analytiquement. Dans ce cas, nous utilisons la méthode dite des quantiles empiriques.

4.1 Simulation de variables aléatoires uniformes indépendantes

Nous avons déjà traité ce problème dans la première partie de ce cours. Dans cette section, nous voulons juste construire un générateur SOBOL afin de mieux comparer graphiquement les simulations. Les procédures suivantes permettent de simuler des nombres aléatoires uniformes et gaussiens à partir du générateur classique à congruence linéaire – `rndCopulaSobol(0)` – ou à partir du générateur SOBOL donné dans Press [23, Teukolsky, Vetterling et Flannery (1992)] – `rndCopulaSobol(dim)`.

Code GAUSS 4.1 – *Générateurs LCG et SOBOL de nombres aléatoires uniformes et gaussiens* –

```
/*
**> rndCopulaSobol
**
*/

proc (0) = rndCopulaSobol(dim);
  local x;
  if dim <= 0;
```

```
    _CopulaSobol = 0;
    retp;
  endif;
  _CopulaSobol = dim;
  dim = -1; x = 0;
  dllcall _dllsobol(dim,x);
  retp;
endp;
```

```
/*
**> rnduCopula
**
*/

proc (1) = rnduCopula(r,c);
  local dim,x,u,i;
  dim = _CopulaSobol;
  if dim /= 0;
    u = zeros(c,r);
    x = zeros(dim,1);
    for i(1,r,1);
      dllcall _dllsobol(dim,x);
      u[.,i] = x[1:c];
    endfor;
    retp( u' );
  else;
    retp( rndu(r,c) );
  endif;
endp;
```

```
/*
**> rndnCopula
**
*/

proc (1) = rndnCopula(r,c);
  if _CopulaSobol /= 0;
    retp( cdfni(rnduCopula(r,c)) );
  else;
    retp( rndn(r,c) );
  endif;
endp;
```

4.2 La méthode des distributions

C'est la méthode inverse de celle présentée dans l'introduction. Nous avons

$$\mathbf{C}\left(U_1,\ldots,U_n\right)=\mathbf{F}\left(\mathbf{F}_1^{-1}\left(U_1\right),\ldots,\mathbf{F}_n^{-1}\left(U_n\right)\right)$$

Pour simuler $\mathbf{U}=(U_1,\ldots,U_n)$, nous pouvons simuler le vecteur aléatoire $\mathbf{X}=(X_1,\ldots,X_n)$ de distribution $\mathbf{F}$ et appliquer la transformation $\mathbf{U}=(\mathbf{F}_1\left(X_1\right),\ldots,\mathbf{F}_n\left(X_n\right))$.

Cette méthode est intéressante si la distribution $\mathbf{F}$ (ou plus exactement si une distribution $\mathbf{F}$ générée par la copule $\mathbf{C}$) est plus facile à simuler que la copule $\mathbf{C}$. C'est par exemple le cas de la copule Normale, puisque nous savons très facilement simuler une distribution normale multidimensionnelle $\mathcal{N}\left(0,\rho\right)$. En effet, nous avons

$$\mathcal{N}\left(0,\rho\right)=\mathbf{P}\mathcal{N}\left(0,\mathbf{I}\right) \tag{4.1}$$

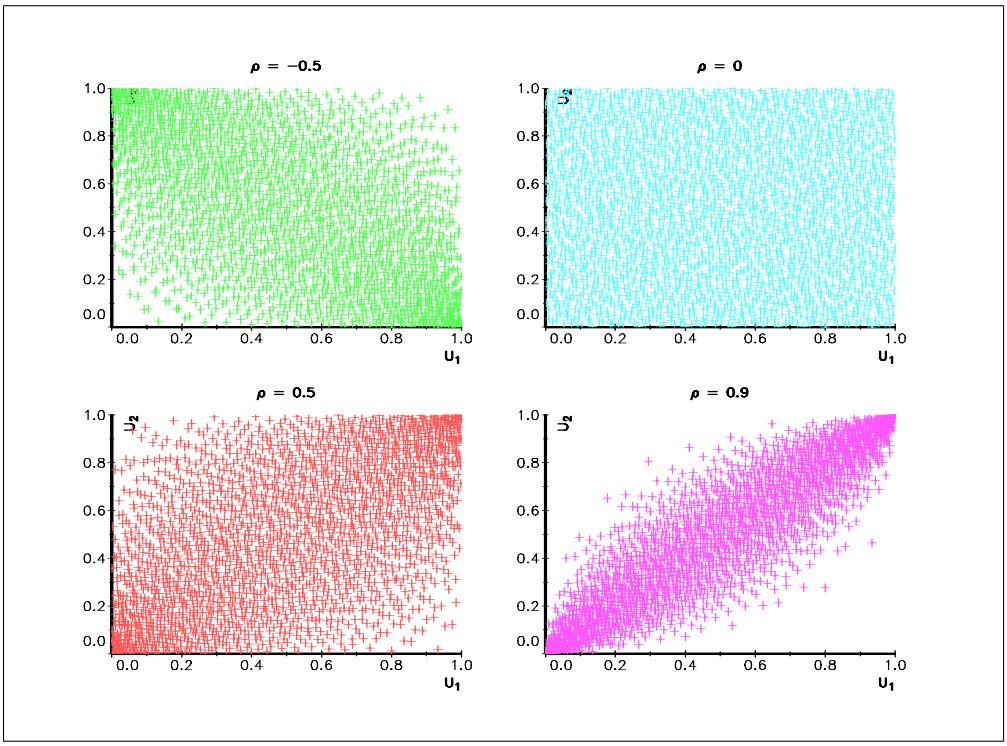
avec $\mathbf{P}$ la décomposition triangulaire inférieure de Cholesky qui vérifie $\mathbf{P}\mathbf{P}^{\top}=\rho$.

Code GAUSS 4.2 – *Simulation de la copule Normale* –

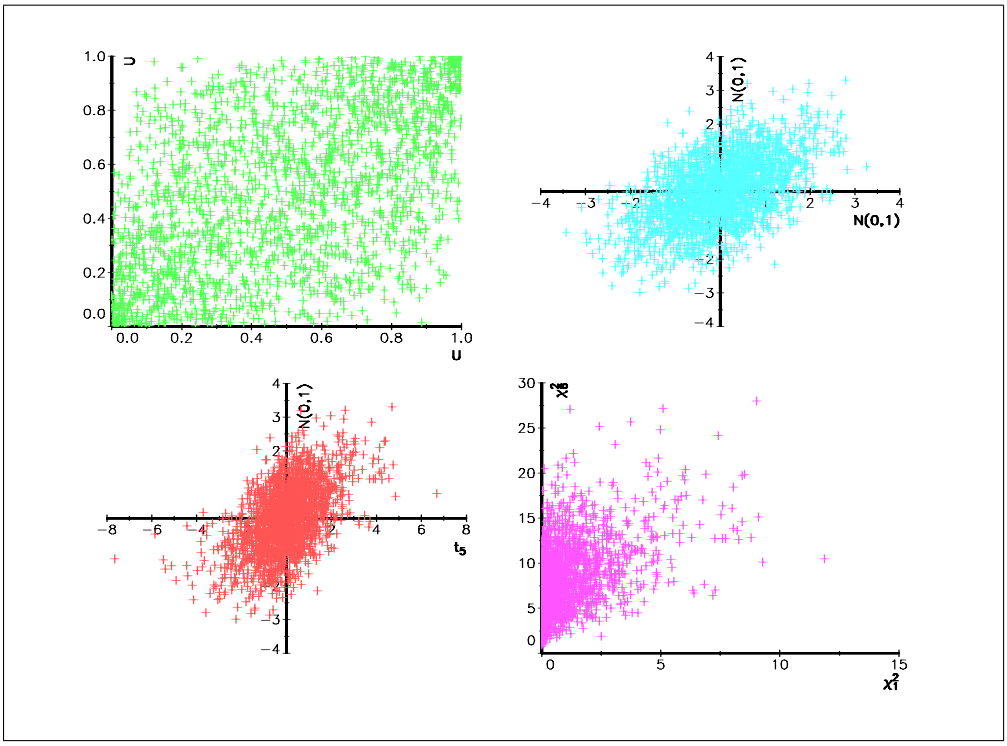
```
/*
**> rndCopulaNormal
**
*/

proc (1) = rndCopulaNormal(rho,ns);
  local u;
  u = rndnCopula(ns,rows(rho));
  retp( cdfn(u*chol(rho)) );
endp;
```

```
/*
**> rndCopulaNormal2
**
*/

proc (2) = rndCopulaNormal2(rho,ns);
  local u;
  u = rndnCopula(ns,2);
  retp( cdfn(u[.,1].*ones(rows(rho),cols(rho))),
        cdfn(rho.*u[.,1]+sqrt(1-rho^2).*u[.,2]) );
endp;
```

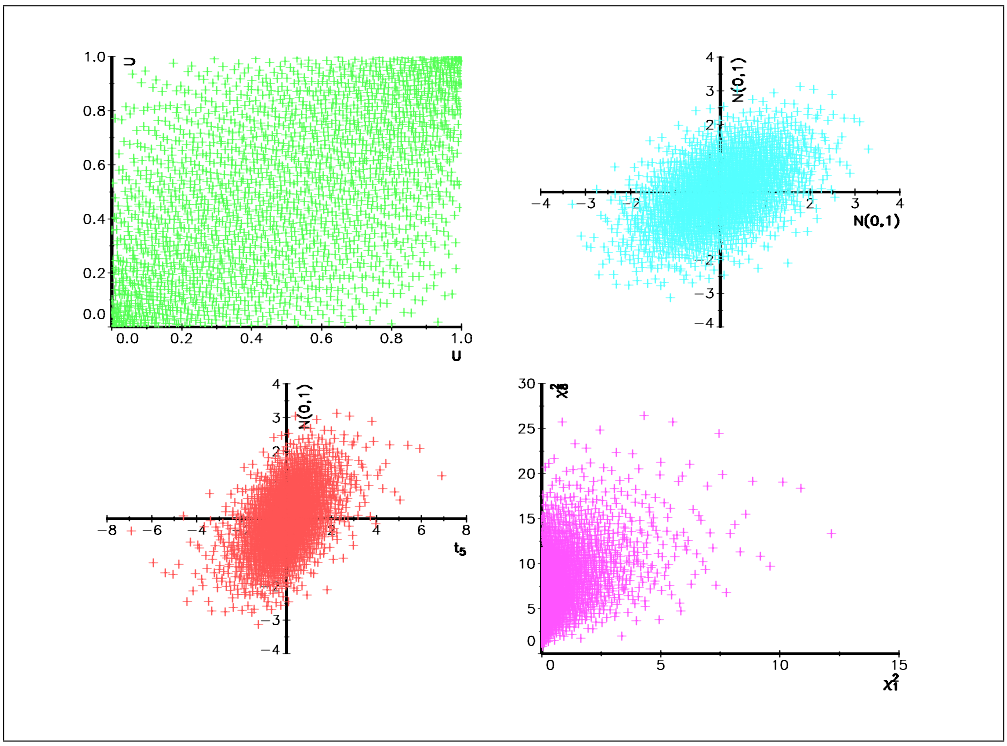
A titre d'illustration, nous simulons 4 distributions $\mathbf{F}(x_1, x_2) = \mathbf{C}(\mathbf{F}_1(x_1), \mathbf{F}_2(x_2))$ où $\mathbf{C}$ est la copule Normale de paramètre $\rho = 0.5$. Les marges sont respectivement uniformes $U_{[0,1]}$, gaussiennes Φ, Student $\mathbf{t}_5$ et gaussienne Φ et chi-deux χ_1^2 et χ_8^2. Pour le graphique 4.2 page suivante, nous utilisons le générateur LCG alors que pour le graphique 4.3 page suivante, nous employons le générateur SOBOL. Nous voyons que le second graphique est plus 'lisible' (du fait que les répartitions sont plus uniformes) que le premier.



Graphique 4.1. Simulation de la copule Normale



Graphique 4.2. Simulation de 4 distributions avec une copule Normale (générateur LCG)



Graphique 4.3. Simulation de 4 distributions avec une copule Normale (générateur SOBOL)

Considérons maintenant la copule Student. Soit $\mathbf{X}$ un vecteur aléatoire dont la distribution est $\mathbf{t}_{\rho,\nu}$. Nous avons

$$\mathbf{X} = \frac{\mathbf{X}^\star}{\sqrt{\chi^2_\nu/\nu}}$$

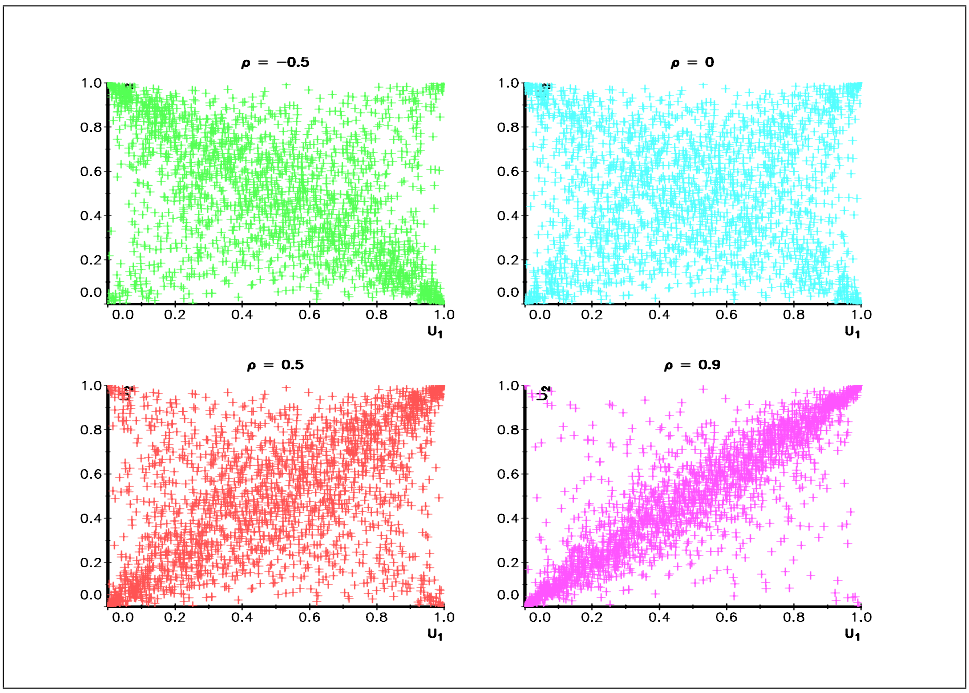
avec $\mathbf{X}^\star$ un vecteur aléatoire gaussien de covariance ρ et χ^2_ν une variable aléatoire chi-deux à ν degrés de libertés. Nous exploitons cette relation pour simuler cette copule.

Code GAUSS 4.3 – *Simulation de la copule Student* –

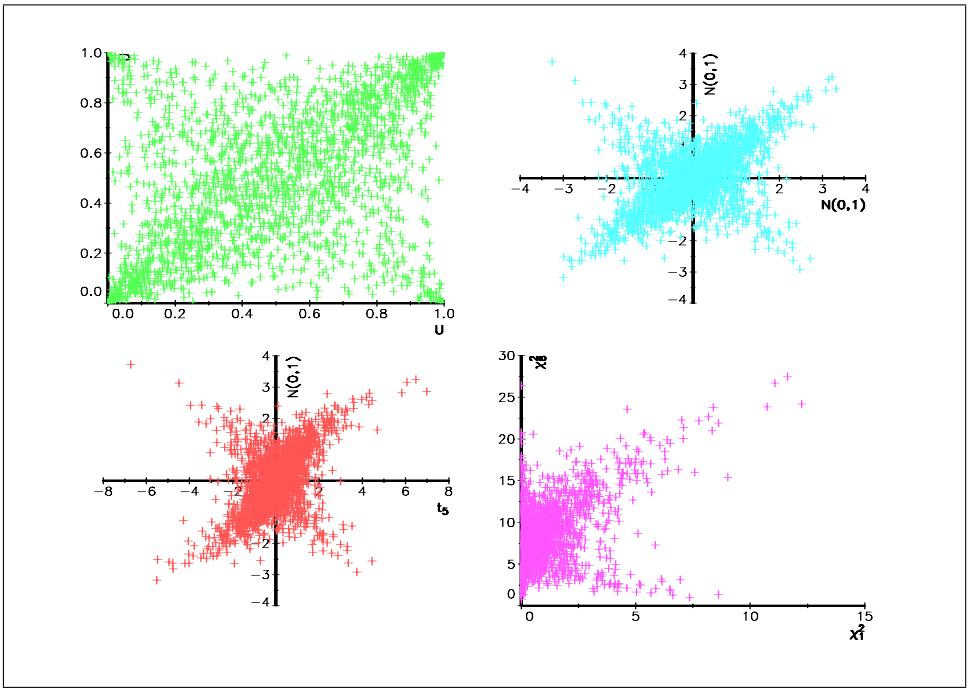
```
/*
**> rndCopulaStudent
**
*/

proc (1) = rndCopulaStudent(rho,nu,ns);
  local u,n,chi2;
  if _CopulaSobol /= 0;
    u = rnduCopula(ns,1+rows(rho));
    n = cdfni(u[.,1:rows(rho)]);
    chi2 = cdfchii(u[.,1+rows(rho)],nu.*ones(ns,1));
  else;
    n = rndn(ns,rows(rho));
    chi2 = cdfchii(rndu(ns,1),nu.*ones(ns,1));
  endif;

  retp( cdft((n*chol(rho))./sqrt(chi2./nu),nu) );
endp;

/*
**> rndCopulaStudent2
**
*/

proc (2) = rndCopulaStudent2(rho,nu,ns);
  local u,n,chi2,n1,n2;
  if _CopulaSobol /= 0;
    u = rnduCopula(ns,3);
    n = cdfni(u[.,1:2]);
    chi2 = cdfchii(u[.,3],nu.*ones(ns,1));
  else;
    n = rndn(ns,2);
    chi2 = cdfchii(rndu(ns,1),nu.*ones(ns,1));
  endif;

  n1 = n[.,1].*ones(rows(rho),cols(rho));
  n2 = rho.*n[.,1]+sqrt(1-rho^2).*n[.,2];
  chi2 = sqrt(chi2./nu);

  retp( cdft(n1./chi2,nu),cdft(n2./chi2,nu) );
endp;
```

A titre d'illustration, nous reprenons les exemples de la copule Normale en utilisant désormais la copule Student (graphiques 4.4 et 4.5). Le graphique 4.6 permet de comparer directement les

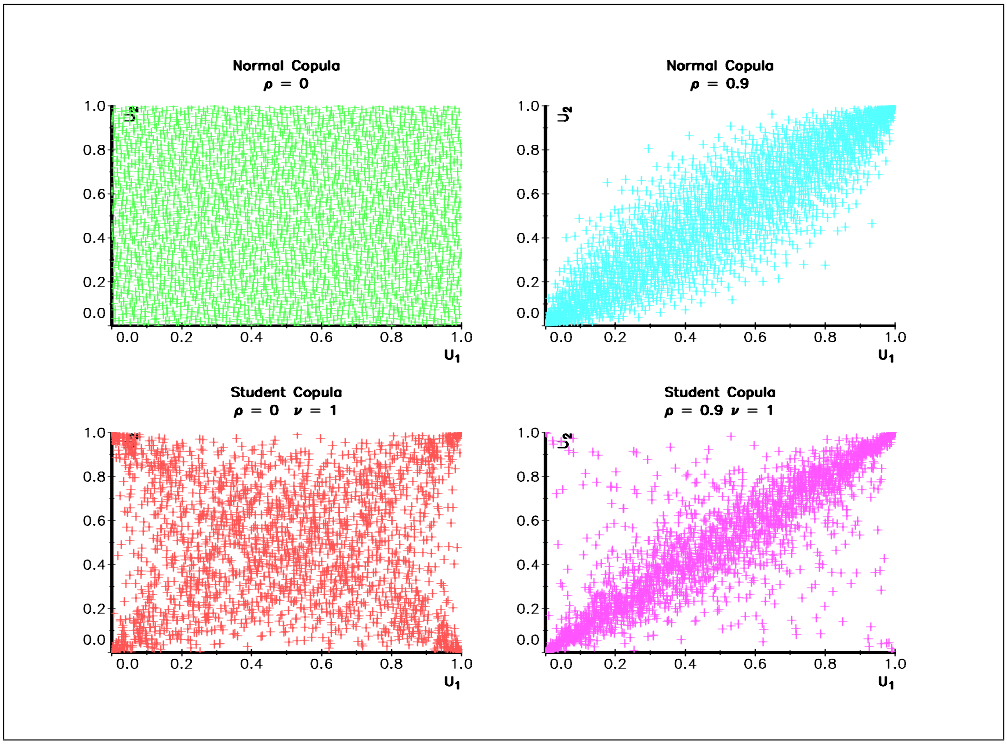
deux copules. Nous vérifions bien que la copule Student corrèle les valeurs extrêmes et qu'elle ne correspond pas à $\mathbf{C}^{\perp}$ lorsque $\rho = 0$.



Graphique 4.4. Simulation de la copule Student ($\nu = 1$)



Graphique 4.5. Simulation de 4 distributions avec une copule Student ($\rho = 0.5$, $\nu = 1$)



Graphique 4.6. Comparaison des copules Normale et Student

4.3 La méthode des distributions conditionnelles

Considérons le cas bivarié. Soit $\mathbf{U} = (U_1, U_2)$ un vecteur aléatoire dont la distribution est $\mathbf{C}$. Nous savons que

$$\mathbb{P}\{U_1 \leq u_1\} = u_1$$

et

$$\mathbb{P}\{U_2 \leq u_2 \mid U_1 = u_1\} = \mathbf{C}_{2|1}(u_1, u_2)$$

Comme $\mathbf{C}(U_1, 1)$ et $\mathbf{C}_{2|1}(u_1, U_2)$ sont deux variables aléatoires uniformes, nous obtenons l'algorithme suivant :

1. Simuler deux variables aléatoires uniformes v_1 et v_2.
2. Prendre u_1 égal à v_1.
3. Soit $\mathbf{C}(u_2; u_1) = \mathbf{C}_{2|1}(u_1, u_2)$. Prendre u_2 égal à $\mathbf{C}^{-1}(v_2; u_1)$.

Cet algorithme est suggéré par Genest et MacKay [11, 1986]. Il est utilisé par Genest [9, 1987] pour simuler la copule Frank. Nous rappelons que la distribution de cette copule est

$$\mathbf{C}(u_1, u_2; \theta) = -\frac{1}{\theta} \ln\left(1 + \frac{\left(e^{-\theta u_1} - 1\right)\left(e^{-\theta u_2} - 1\right)}{e^{-\theta} - 1}\right).$$

Nous en déduisons que

$$\mathbf{C}_{2|1}(u_1, u_2; \theta) = \frac{\left(e^{-\theta u_2} - 1\right) e^{-\theta u_1}}{\left(e^{-\theta} - 1\right) + \left(e^{-\theta u_1} - 1\right)\left(e^{-\theta u_2} - 1\right)}.$$

Finalement, nous obtenons

$$\begin{aligned}\mathbf{C}^{-1}(u;u_1) &= \left\{u_2 : \mathbf{C}_{2|1}(u_1,u_2;\theta) = u\right\} \\ &= -\frac{1}{\theta}\ln\left(1+\frac{u\left(e^{-\theta}-1\right)}{u+(1-u)\,e^{-\theta u_1}}\right).\end{aligned}$$

Code GAUSS 4.4 – *Simulation de la copule Frank* –

```
/*
**> rndCopulaFrank
**
*/

proc (2) = rndCopulaFrank(theta,ns);
  local v,u1,u2;
  v = rnduCopula(ns,2);
  u1 = v[.,1] .* ones(rows(theta),cols(theta));
  u2 = -ln(1 + v[.,2].*(exp(-theta)-1)./(v[.,2] + (1-v[.,2])
       .*exp(-theta.*u1) ))./theta;
  retp(u1,u2);
endp;
```

Dans de nombreux cas, il n'est pas possible d'obtenir une formule analytique de $\mathbf{C}^{-1}(u;u_1)$ (voir [15, Joe (1987), pages 146 et 147]). Dans ce cas, nous pouvons résoudre l'équation $\mathbf{C}_{2|1}(u_1,u_2;\theta) = u$ par une méthode numérique. La procédure suivante `rndCopula2` est construite à partir d'une bisection classique. Lorsque la distribution conditionnelle $\mathbf{C}_{2|1}(u_1,u_2;\theta)$ n'est pas spécifiée, celle-ci est obtenue par une méthode de gradient numérique.

Code GAUSS 4.5 – *Simulation d'une copule bivariée par la méthode des distributions conditionnelles* –

```
/*
**> rndCopula2
**
*/

proc (2) = rndCopula2(cdfCopula,
                      cndCopula,ns);
  local v,u1,u2;
  _CopulaFunction = cdfCopula|cndCopula;
  v = rnduCopula(ns,2);
  _Copula_v = v;
  u1 = v[.,1];
  u2 = rndCopulaFindZero(&_rndCopula2,
                         0+__macheps,
                         1-__macheps);
  retp(u1,u2);
endp;
```

```
proc _rndCopula2(u2);
  local cdfCopula,cndCopula,u,w;
  cdfCopula = _CopulaFunction[1];
  cndCopula = _CopulaFunction[2];
  if cndCopula /= 0;
    local cndCopula:proc;
    u = cndCopula(_Copula_v[.,1],u2);
  else;
    {u,w} = gradp2D(cdfCopula,
                    _Copula_v[.,1],u2);
  endif;
  retp( u - _Copula_v[.,2] );
endp;
```

```
proc rndCopulaFindZero(f,a,b);
  local f:proc;
  local ya,yb,Nobs,c,yc,indx1,indx2;
  local indx,s,diff,const;

  ya = f(a); yb = f(b);
  Nobs = rows(ya);
  a = a .* ones(Nobs,1);
  b = b .* ones(Nobs,1);

  indx = (ya .< 0) .and (yb .> 0);
  if sumc(indx) == 0;
    retp(miss(zeros(Nobs,1),0));
  endif;

  if sumc(indx) == Nobs;

    do while maxc(abs(a-b)) > _rndCopulaFindZero_Tol;
      c = (a+b)/2;
      yc = f(c);
      indx1 = yc.<0;
      indx2 = 1 - indx1;
      a = indx1.*c + indx2.*a;
      b = indx1.*b + indx2.*c;
    endo;

  else;

    s = delif(seqa(1,1,Nobs),indx);
    diff = selif(a-b,indx);
    const = miss(zeros(Nobs-sumc(indx),1),0);

    do while maxc(abs(diff)) > _rndCopulaFindZero_Tol;
      c = (a+b)/2;
      c[s] = const;
      yc = f(c);
      indx1 = yc.<0;
      indx2 = 1 - indx1;
      a = indx1.*c + indx2.*a;
      b = indx1.*b + indx2.*c;
      diff = selif(a-b,indx);
    endo;

  endif;

  c = (a+b)/2;

  retp(c);
endp;
```

A partir de cette procédure, nous pouvons très facilement simuler la copule Gumbel soit en utilisant la distribution bivariée soit en utilisant la distribution conditionnelle. Rappelons que la

copule Gumbel (ou Gumbel-Houggaard) a pour expression :

$$\mathbf{C}\left(u_1, u_2; \theta\right) = \exp\left(-\left[\left(-\ln u_1\right)^{\theta} + \left(-\ln u_2\right)^{\theta}\right]^{1/\theta}\right).$$

La distribution conditionnelle s'écrit donc :

$$\mathbf{C}_{2|1}\left(u_1, u_2\right) = \frac{1}{u_1}\left[1 + \left(\frac{\ln u_2}{\ln u_1}\right)^{\theta}\right]^{\frac{1}{\theta}-1} \exp\left(-\left[\left(-\ln u_1\right)^{\theta} + \left(-\ln u_2\right)^{\theta}\right]^{1/\theta}\right)$$

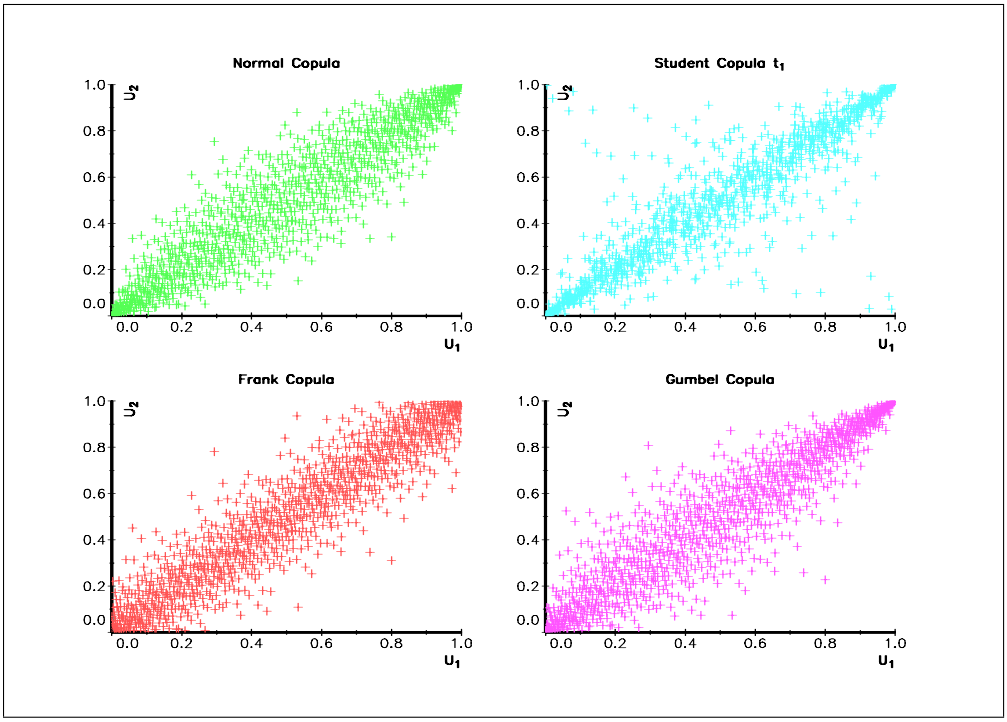
Code GAUSS 4.6 – *Simulation de la copule Gumbel à partir de la distribution bivariée* –

```
/*
**> rndCopulaGumbel
**
*/

proc (2) = rndCopulaGumbel(theta,ns);
  _CopulaParameters = theta;
  retp( rndCopula2(&_rndCopulaGumbel,0,ns) );
endp;

proc _rndCopulaGumbel(u1,u2);
  retp( _cdfCopulaGumbel(u1,u2,_CopulaParameters) );
endp;

proc _cdfCopulaGumbel(u1,u2,theta);
  retp( exp(-((-ln(u1))^theta+(-ln(u2))^theta)^(1./theta)) );
endp;
```

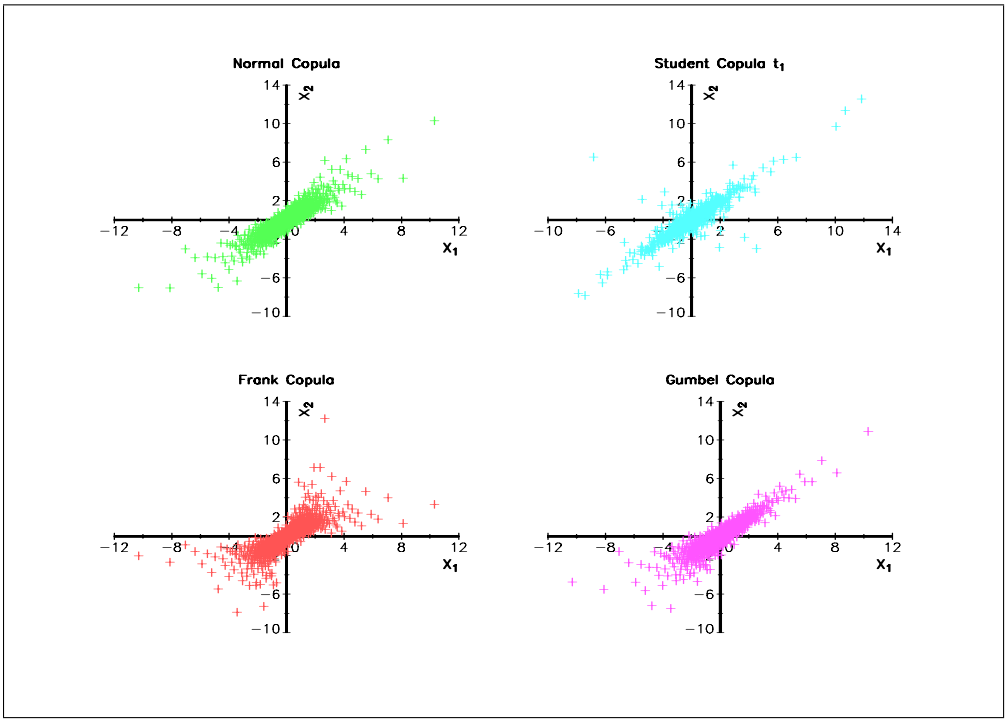
Code GAUSS 4.7 – *Simulation de la copule Gumbel à partir de la distribution conditionnelle* –

```
/*
**> rndCopulaGumbel
**
*/

proc (2) = rndCopulaGumbel(theta,ns);
  _CopulaParameters = theta;
  retp( rndCopula2(0,&_rndCopulaGumbel,ns) );
endp;

proc _rndCopulaGumbel(u1,u2);
  local theta,u1tilde,u2tilde,beta,w;
  theta = _CopulaParameters;
  u1tilde = -ln(u1); u2tilde = -ln(u2);
  w = u1tilde^theta + u2tilde^theta;
  beta = 1./theta;
  retp( exp(-(w^beta)) .* ( 1+ (u2tilde./u1tilde)^theta )^(-1+beta) ./ u1 );
endp;
```

Bien sûr, la seconde version est plus stable, puisqu'elle ne fait pas appel à une dérivation numérique. Cependant, les deux procédures donnent pratiquement les mêmes nombres aléatoires, puisque la différence maximale sur les 1024 premiers nombres du générateur Sobol est 7.57×10^{-6}.



Graphique 4.7. Simulation des copules Normale, Student, Frank et Gumbel (marges uniformes)



Graphique 4.8. Simulation des copules Normale, Student, Frank et Gumbel (marges $\mathbf{t}_3$)

La méthode bivariée s'étend sans difficulté (sur le plan mathématique) au cas multivarié. Prenons par exemple le cas trivarié. Nous avons l'algorithme suivant :

1. Simuler trois variables aléatoires uniformes v_1, v_2 et v_3.
2. Prendre u_1 égal à v_1.
3. Soit $\mathbf{C}(u_2; u_1) = \mathbf{C}_{2|1}(u_1, u_2, 1)$. Prendre u_2 égal à $\mathbf{C}^{-1}(v_2; u_1)$.
4. Soit $\mathbf{C}(u_3; u_1, u_2) = \mathbf{C}_{3|1,2}(u_1, u_2, u_3)$. Prendre u_3 égal à $\mathbf{C}^{-1}(v_3; u_1, u_2)$.

La seule difficulté **numérique** est donc le calcul de la distribution conditionnelle $\mathbf{C}_{m|1,\ldots,m-1}(u_1, \ldots, u_n)$ pour $m \leq n$. Cela n'est cependant pas un problème si l'on maîtrise la récursivité.

4.4 Les méthodes dites analytiques

Dans ce paragraphe, la simulation est spécifique à chaque copule ou à un type de copule.

Par exemple, pour simuler la copule Clayton (pour $\theta > 0$), nous pouvons employer l'algorithme donné par Devroye [6, 1986] :

1. Simuler deux variables aléatoires exponentielles standards x_1 et x_2.
2. Simuler une variable aléatoire x de distribution $\Gamma(1, \theta)$.
3. Prendre $u_1 = (1 + x_1/x)^{-\theta}$ et $u_2 = (1 + x_2/x)^{-\theta}$.

Code GAUSS 4.8 – *Simulation de la copule Clayton* –

```
/*
**> rndCopulaClayton
**
*/

proc (2) = rndCopulaClayton(theta,ns);
  local x1,x2,x,u1,u2;
  x1 = -ln(rndu(ns,1));
  x2 = -ln(rndu(ns,1));
  x = rndgam(ns,1,theta);
  u1 = (1+x1./x)^(-theta);
  u2 = (1+x2./x)^(-theta);
  retp(u1,u2);
endp;
```

Il existe plusieurs algorithmes pour simuler les copules Archimédiennes[1]. Genest et MacKay [10, 1986] proposent celui-ci :

1. Simuler deux variables aléatoires uniformes v_1 et v_2.
2. Prendre $u_1 = v_1$.
3. Prendre $u_2 = \varphi^{-1}\left(\varphi\left(\varphi'^{-1}\left(\varphi'\left(\frac{v_1}{v_2}\right)\right)\right) - \varphi(v_1)\right)$.

Le lecteur pourra montrer en exercice qu'il s'agit de l'algorithme des distributions conditionnelles.

[1] Genest et MacKay [10, 1986] définissent une copule Archimédienne (Archimedean copula) de la façon suivante :

$$\mathbf{C}(u_1, u_2) = \begin{cases} \varphi^{-1}(\varphi(u_1) + \varphi(u_2)) & \text{si } \varphi(u_1) + \varphi(u_2) \leq \varphi(0) \\ 0 & \text{sinon} \end{cases}$$

avec φ une fonction de classe C^2 qui vérifie $\varphi(1) = 0$, $\varphi'(u) < 0$ et $\varphi''(u) > 0$ pour tout $u \in [0, 1]$. $\varphi(u)$ est appelé le *générateur* (ou la fonction génératrice) de la copule. Ce générateur est dit *strict* si $\varphi(0) = \infty$ et non-strict si $\varphi(0) < \infty$.

4.5 La méthode des quantiles empiriques

Le problème de cette section n'est plus la simulation du vecteur $\mathbf{U}$ dont la distribution est une copule $\mathbf{C}$, mais concerne la simulation du vecteur $\mathbf{X}$ dont la copule est $\mathbf{C}$ et les marges pas forcément uniformes. Dans les sections précédentes, $\mathbf{X}$ est simulé à partir de la transformation suivante :

$$\mathbf{X} = \begin{pmatrix} \mathbf{F}_1^{-1}(U_1) \\ \vdots \\ \mathbf{F}_n^{-1}(U_n) \end{pmatrix}$$

Cela implique de connaître les distributions analytiques $\mathbf{F}_1, \ldots, \mathbf{F}_n$. Cela n'est pas toujours le cas (pensez au risque opérationnel et aux distributions composées des pertes, ou encore à un modèle bidimensionnel de volatilité stochastique à la Heston). Néanmoins, s'il est possible de simuler les marges $\mathbf{F}_1, \ldots, \mathbf{F}_n$, alors nous pouvons simuler la distribution multidimensionnelle $\mathbf{F}$ grâce à la méthode des quantiles empiriques. Soit $\mathbb{F}_{i,m}$ le processus de distribution empirique (non normalisé). Nous avons le résultat suivant [24, Shorack et Wellner,1986] :

$$\sup_x |\mathbb{F}_{i,m}(x) - \mathbf{F}_i(x)| \overset{\text{a.s.}}{\to} 0 \text{ lorsque } m \to 0$$

Soient $\mathbb{U}_m$ et $\mathbb{F}_m$ les processus de distribution empirique correspondants aux distributions $\mathbf{C}(u_1, \ldots, u_n)$ et $\mathbf{F}(x_1, \ldots, x_n)$. En utilisant un argument de type Glivenko-Cantelli, nous avons

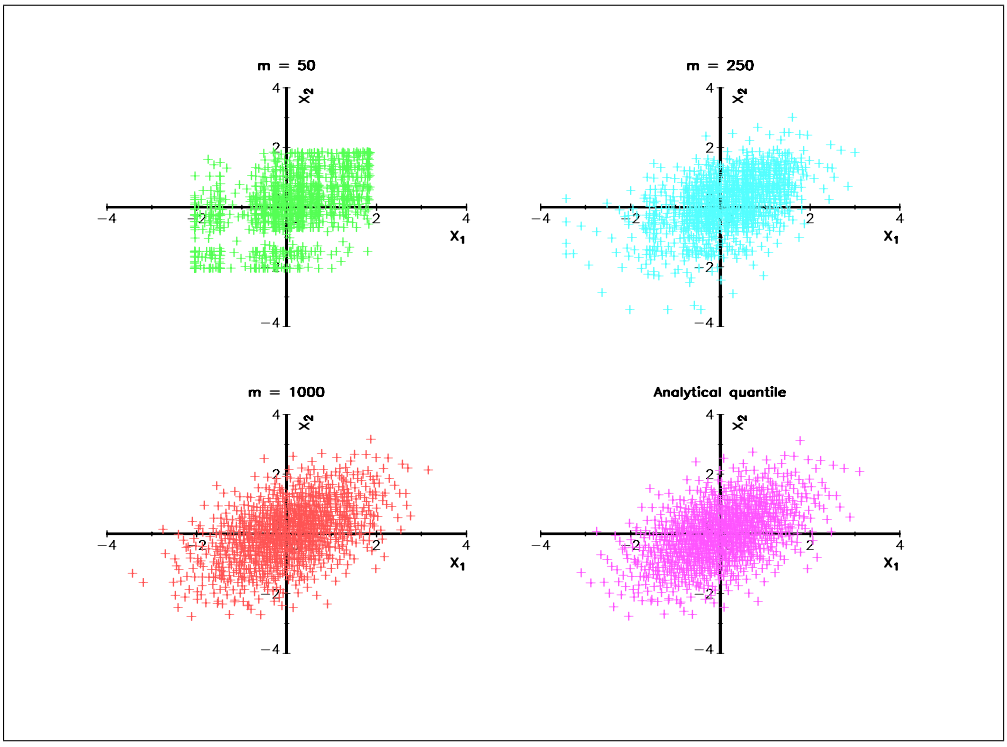
$$\sup_{u_1,\ldots,u_n} \left|\mathbb{U}_p\left(\mathbb{F}_{1,m}^{-1}(u_1), \ldots, \mathbb{F}_{n,m}^{-1}(u_n)\right) - \mathbf{C}\left(\mathbf{F}_1^{-1}(u_1), \ldots, \mathbf{F}_n^{-1}(u_n)\right)\right| \overset{\text{a.s.}}{\to} 0 \text{ lorsque } m \wedge p \to 0$$

Code GAUSS 4.9 – *Simulation de la distribution multidimensionnelle* $\mathbf{F}$ *par la méthode des quantiles empiriques* –

```
/*
**> rndCopulaEmpiricalQuantile
**
*/

proc (1) = rndCopulaEmpiricalQuantile(u,x);
  local n,ns,y,i;

  n = cols(u);
  ns = rows(u);
  y = zeros(ns,n);
  x = x .* ones(1,n);

  i = 1;
  do until i > n;
    y[.,i] = _rndCopulaEmpiricalQuantile(x[.,i],u[.,i]);
    i = i + 1;
  endo;

  retp(y);
endp;
```

```
proc _rndCopulaEmpiricalQuantile(x,e);
  local w,wt,f,z,r;
  w = rows(x) * e;
  wt = floor(w);
  f = w - wt;
  z = sortc(x,1);
  z = z[1]|z|z[rows(x)];
  wt = wt + 1;
  r = z[wt,.] + f .* (z[wt+1,.] - z[wt,.]);
  retp(r);
endp;
```

A titre d'illustration, le graphique 4.8 montre la convergence de la méthode des quantiles empiriques vers la méthode des distributions. X_1 et X_2 sont deux variables aléatoires gaussiennes (copule Normale de paramètre 0.5). m est le nombre de simulations pour construire les fonctions $\mathbb{F}_1(x_1)$ et $\mathbb{F}_2(x_2)$.



Graphique 4.9. Convergence de la méthode des quantiles empiriques vers la méthode des distributions

4.6 Repères historiques

Il existe peu d'articles sur le sujet. Vous trouverez des compléments dans [6, Devroye,1986] et surtout [16, Johnson,1987] (la plupart de ces algorithmes sont aussi dans [19, Nelsen,1999]).

5

Méthodes de Monte Carlo appliquées à le gestion des risques

5.1 Le problème de l'agrégation de (et dans) la valeur en risque

Considérons deux variables aléatoires X_1 et X_2 représentant les pertes de deux marchés. Soient $\mathbf{F}_1$ et $\mathbf{F}_2$ les distributions de X_1 et X_2. Les valeurs en risque au seuil de confiance α sont alors

$$VaR\left(X_1;\alpha\right)=\mathbf{F}_1^{-1}\left(\alpha\right)$$

et

$$VaR\left(X_2;\alpha\right)=\mathbf{F}_2^{-1}\left(\alpha\right)$$

Nous cherchons à agréger ces deux valeurs en risque, c'est-à-dire à déterminer la VaR de X_1+X_2 :

$$VaR\left(X_1+X_2;\alpha\right):=\inf\left\{x:\mathbb{P}\left\{X_1+X_2\leq x\right\}\geq\alpha\right\}$$

Soit $\mathbf{F}_{1+2}$ la distribution de X_1+X_2. Nous avons

$$\mathbf{F}_{1+2}\left(x\right)=\iint_{x_1+x_2\leq x}\mathrm{d}\mathbf{C}\left(\mathbf{F}_1\left(x_1\right),\mathbf{F}_2\left(x_2\right)\right)$$

La valeur en risque $VaR\left(X_1+X_2;\alpha\right)$ dépend des distributions marginales, mais aussi de la copule $\mathbf{C}$.

Agrégation sous l'hypothèse $\mathbf{C}=\mathbf{C}^+$

Nous considérons le cas $\mathbf{C}=\mathbf{C}^+$ et nous supposons que $\mathbf{F}_1$ et $\mathbf{F}_2$ sont continues. Nous avons $X_2=\mathbf{F}_2^{-1}\left(\mathbf{F}_1\left(X_1\right)\right)$. Soit la fonction ϖ définie par $x\mapsto x+\mathbf{F}_2^{-1}\left(\mathbf{F}_1\left(x\right)\right)$. Nous avons

$$\begin{aligned}\alpha &= \mathbb{P}\left\{X_1+X_2\leq VaR\left(X_1+X_2;\alpha\right)\right\}\\ &= \mathbb{E}\left[\mathbf{1}\left\{\varpi\left(X_1\right)\leq Var\left(X_1+X_2;\alpha\right)\right\}\right]\\ &= \mathbf{F}_1\left(\varpi^{-1}\left(Var\left(X_1+X_2;\alpha\right)\right)\right)\end{aligned}\tag{5.1}$$

Nous en déduisons que $Var\left(X_1+X_2;\alpha\right)=\varpi\left(\mathbf{F}_1^{-1}\left(\alpha\right)\right)$ et nous obtenons le résultat suivant

$$\begin{aligned} Var\left(X_1+X_2;\alpha\right) &= \mathbf{F}_1^{-1}\left(\alpha\right)+\mathbf{F}_2^{-1}\left(\mathbf{F}_1\left(\mathbf{F}_1^{-1}\left(\alpha\right)\right)\right) \\ &= \mathbf{F}_1^{-1}\left(\alpha\right)+\mathbf{F}_2^{-1}\left(\alpha\right) \\ &= Var\left(X_1;\alpha\right)+Var\left(X_2;\alpha\right) \end{aligned}$$

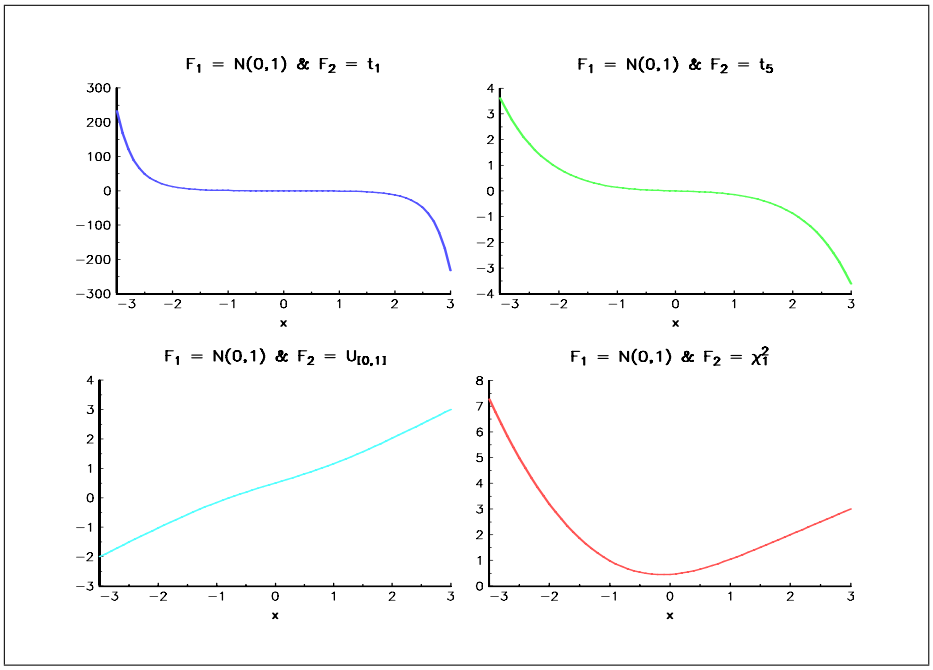
Théorème 5.1 *Le principe d'agrégation des valeurs en risque qui consiste à sommer les VaRs individuelles repose sur l'hypothèse que la dépendance entre les pertes aléatoires est la copule Fréchet* $\mathbf{C}^+$.

Agrégation sous l'hypothèse $\mathbf{C}=\mathbf{C}^-$

Nous considérons maintenant le cas $\mathbf{C}=\mathbf{C}^-$. Nous avons $X_2=\mathbf{F}_2^{-1}\left(1-\mathbf{F}_1\left(X_1\right)\right)$. Soit la fonction ϖ définie par $x\mapsto x+\mathbf{F}_2^{-1}\left(1-\mathbf{F}_1\left(x\right)\right)$. Nous avons

$$\begin{aligned} \alpha &= \mathbb{P}\left\{X_1+X_2\leq Var\left(X_1+X_2;\alpha\right)\right\} \\ &= \mathbb{E}\left[\mathbf{1}\left\{\varpi\left(X_1\right)\leq Var\left(X_1+X_2;\alpha\right)\right\}\right] \end{aligned}$$

Contrairement au cas précédent, ϖ n'est plus forcément une fonction monotone croissante. A titre d'exemple, nous représentons la fonction $\varpi\left(x\right)$ sur le graphique 5.1 pour différentes fonctions $\mathbf{F}_1$ et $\mathbf{F}_2$. $\varpi\left(x\right)$ peut donc être croissante, décroissante ou non monotone. Par exemple, dans



Graphique 5.1. Fonction $\varpi\left(x\right)$ pour différentes fonctions $\mathbf{F}_1$et $\mathbf{F}_2$

le cas croissant, nous avons $Var\left(X_1+X_2;\alpha\right)=\varpi\left(\mathbf{F}_1^{-1}\left(\alpha\right)\right)$ et nous obtenons $Var\left(X_1+X_2;\alpha\right)=Var\left(X_1;\alpha\right)+Var\left(X_2;1-\alpha\right)$. Dans les autres cas, nous avons $Var\left(X_1+X_2;\alpha\right)=Var\left(X_1;1-\alpha\right)+Var\left(X_2;\alpha\right)$ ou d'autres expressions plus complexes.

Agrégation dans le cas de la copule Normale

Considérons le cas où $\mathbf{X} = (X_1, X_2)$ est un vecteur aléatoire gaussien standard de corrélation ρ. Nous avons $Var(X_1; \alpha) = Var(X_2; \alpha) = \Phi^{-1}(\alpha)$. $X_1 + X_2$ est une variable aléatoire gaussienne centrée de variance $2(1+\rho)$. Nous en déduisons que

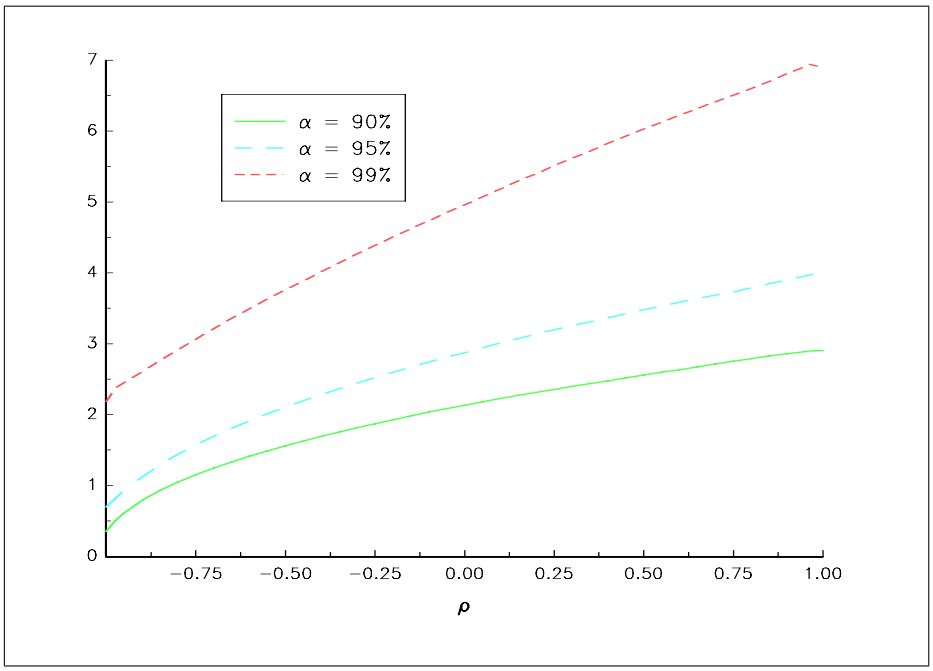
$$Var(X_1 + X_2; \alpha) = \sqrt{2(1+\rho)}\Phi^{-1}(\alpha)$$

Nous avons donc les cas particuliers suivants :

ρ	-1	0	1
$Var(X_1 + X_2)$	0	$[Var(X_1) + Var(X_2)]^{1/2}$	$Var(X_1) + Var(X_2)$

Encore une fois, il apparaît que le choix de la dépendance est primordial pour agréger les risques.

Considérons maintenant le cas où $X_1 \sim \mathcal{N}(0,1)$ et $X_2 \sim \mathbf{t}_3$. Le graphique 5.2 représente $Var(X_1 + X_2; \alpha)$ en fonction du paramètre ρ de la copule Normale.



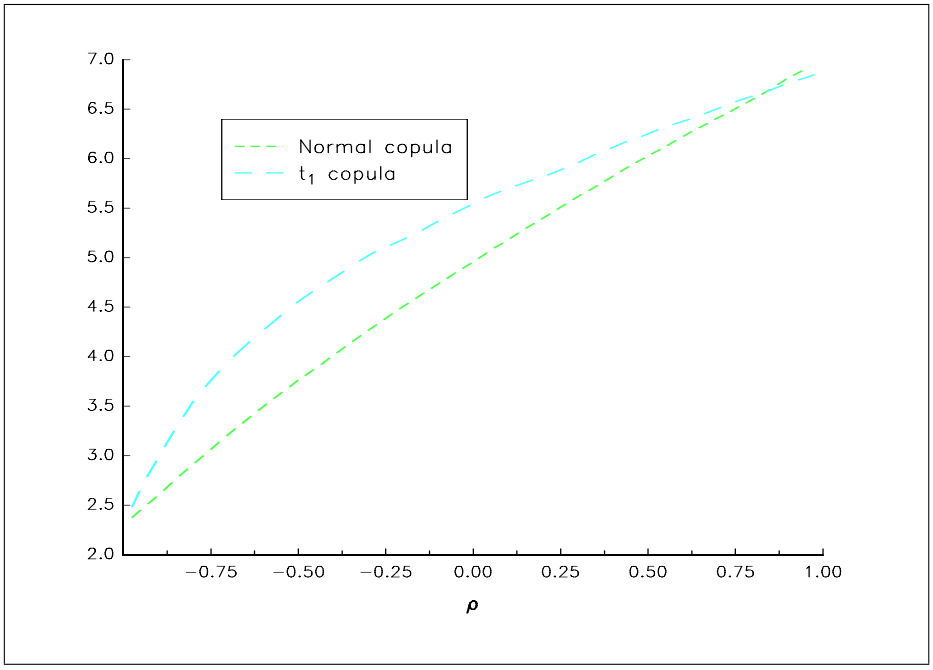
Graphique 5.2. $Var(X_1 + X_2; \alpha)$ lorsque $X_1 \sim \mathcal{N}(0,1)$, $X_2 \sim \mathbf{t}_3$ et $\mathbf{C}\langle X_1, X_2\rangle$ est une copule Normale

Utilisation de la la copule Student

Nous Reprenons ici l'exemple précédent ($X_1 \sim \mathcal{N}(0,1)$ et $X_2 \sim \mathbf{t}_3$) non plus dans le cas de la copule Normale mais dans celui de la copule Student. Le graphique 5.3 page suivante représente $Var(X_1 + X_2; \alpha)$ en fonction du paramètre ρ des copules Normale et Student ($\nu = 1$). Nous constatons que la mesure est plus grande pour cette dernière copule.

Les différentes composantes de la mesure des risques

L'approche copule de la mesure du risque que nous venons de développer dans les exemples précédents met en évidence les deux principales sources de risque à modéliser :



Graphique 5.3. Comparaison de la mesure $Var\left(X_1 + X_2; 99\%\right)$ (copules Normale et $\mathbf{t}_1$)

1. La première 'source de risque' concerne évidemment les facteurs pris de façon individuelle. Il est évident que le choix de modélisation de la distribution d'un facteur peut avoir une influence **importante** sur la mesure de risque. De plus, même si deux distributions ont été calibrées pour donner la même mesure pour un seuil de confiance donné, le passage à un seuil de confiance plus élevé peut entraîner des différences énormes. Bien sûr, ceci est lié au comportement (asymptotique et non-asymptotique) des valeurs extrêmes. Par exemple, la table 5.1 illustre les différences induites par des distributions Φ, $\mathbf{t}_4$ et $\mathbf{t}_2$.

Rating	Réglementaire (marché)	BBB	A	AA	AAA
α	99%	99.75%	99.9%	99.95%	99.97%
Temps de retour	100 jours	400 jours	4 années	8 années	13 années
$\Phi^{-1}(\alpha)$	2.33	2.81	3.09	3.29	3.43
$\mathbf{t}_4^{-1}(\alpha)$	3.75	5.60	7.17	8.61	9.83
$\frac{\Phi^{-1}(0.99)}{\mathbf{t}_4^{-1}(0.99)}\mathbf{t}_4^{-1}(\alpha)$	2.33	3.48	4.45	5.35	6.10
$\mathbf{t}_2^{-1}(\alpha)$	6.96	14.1	22.3	31.6	40.8
$\frac{\Phi^{-1}(0.99)}{\mathbf{t}_2^{-1}(0.99)}\mathbf{t}_2^{-1}(\alpha)$	2.33	4.71	7.46	10.6	13.6

Tableau 5.1. Influence de la distribution sur la mesure de risque

2. La seconde 'source de risque' porte sur la dépendance des facteurs, c'est-à-dire comment sont corrélés les risques unidimensionnels. Il est difficile d'analyser le risque multidimensionnel lorsque nous n'utilisons pas une approche copule, car il n'est pas évident de voir quelle est la part des marginales et la part de la copule. La famille de la copule est un choix important à faire. Nous rappelons que le problème de l'agrégation de la mesure de risque est un problème d'agrégation de quantiles extrêmement élevés ($\alpha > 99\%$). Il est évident qu'une copule qui présente une dépendance de queue ne corrèle pas ces quantiles de la même façon qu'une copule qui n'a pas de dépendance de queue.

5.2 Différentes méthodes de construction la VaR de marché

Nous présentons dans cette section quatre méthodes de calcul de la VaR de marché. Les deux premières approches ne requièrent pas l'utilisation de simulations Monte Carlo : il s'agit de La VaR historique et de la VaR gaussienne.

Les deux approches suivantes sont des approches paramétriques dans lesquelles la dépendance entre les différents facteurs de risque est modélisée au moyen d'une copule. Ces deux approches diffèrent par la façon dont chaque source de risque est prise en compte :

- Dans l'approche paramétrique, la loi de chaque facteur de risque est issue d'un modèle paramétrique ;
- Dans l'approche semi-historique (ou semi-paramétrique), la loi des différents facteurs correspond à la distribution empirique.

Cadre d'analyse

Soit un portefeuille linéaire de n actifs. Nous notons $P_i(t)$ et $r_i(t)$ le prix et le rendement de l'actif i à la date t. Si $\theta = (\theta_1, \ldots, \theta_n)$, la valeur du portefeuille à la date t est

$$P(t) = \sum_{i=1}^{n} \theta_i P_i(t)$$

Nous supposons que les facteurs de risque sont les rendements des actifs. A la date t, la valeur du portefeuille $P(t+1)$ est aléatoire et nous avons

$$\begin{aligned} P(t+1) &= \sum_{i=1}^{n} \theta_i P_i(t+1) \\ &= \sum_{i=1}^{n} \theta_i P_i(t)(1 + r_i(t+1)) \end{aligned}$$

Le P&L entre les dates t et $t+1$ est donc

$$\begin{aligned} \Pi(t+1 \mid t) &= P(t+1) - P(t) \\ &= \sum_{i=1}^{n} \theta_i P_i(t) r_i(t+1) \\ &= \theta_\star(t)^\top \mathbf{r}(t+1) \end{aligned}$$

où $\mathbf{r}(t+1)$ est le vecteur aléatoire des rendements $(r_1(t+1), \ldots, r_n(t+1))$ et $\theta_\star(t)$ est le vecteur dont la i-ième composante est $\theta_i P_i(t)$. Notons $\mathbf{F}$ la distribution de $\mathbf{r}(t+1)$. La valeur en risque au seuil de confiance α est alors définie par

$$\begin{aligned} Var(\alpha) &= -\inf\{x : \mathbb{P}\{\Pi(t+1 \mid t) \leq x\} \geq 1 - \alpha\} \\ &= -\mathbf{F}^{-1}(1-\alpha) \end{aligned}$$

Pour chaque méthode de calcul de la VaR, les applications numériques utilisent la base de données du London Metal Exchange étudiée dans [2, Bouyé et al., 2000]. Nous considérons en particulier la dépendance entre le prix spot de l'aluminium (que nous notons AL) et le prix forward 15 mois (que nous notons AL-15), ainsi que les prix spot du cuivre (CU), du nickel (NI) et du plomb (PB). Les trois portefeuilles considérés ont la composition suivante :

	AL	AL-15	CU	NI	PB
P_1	1	1	1	1	1
P_2	-1	-1	-1	1	1
P_3	2	1	-3	4	5

5.2.1 VaR historique

Pour la VaR historique, $\mathbf{F}$ est modélisée par la distribution empirique $\hat{\mathbf{F}}$. Supposons que nous disposions d'un historique de m observations des rendements. Soit $\hat{\mathbf{r}}^j = \left(\hat{r}_1^j, \ldots, \hat{r}_n^j\right)$ le vecteur des rendements de la j-ième observation. Nous pouvons calculer le P&L historique associé

$$\hat{\Pi}^j = \theta_\star(t)^\top \hat{\mathbf{r}}_j$$

La distribution empirique $\hat{\mathbf{F}}$ est alors caractérisée par le vecteur $\left(\hat{\Pi}^1, \ldots, \hat{\Pi}^m\right)$ et $\hat{\mathbf{F}}^{-1}(1-\alpha)$ correspond au quantile $1-\alpha$ de $\left(\hat{\Pi}^1, \ldots, \hat{\Pi}^m\right)$.

	90%	95%	99%	99.5%	99.9%
P_1	6.627	9.023	14.43	16.52	29.56
P_2	3.434	5.008	8.946	11.28	16.24
P_3	12.24	17.32	31.77	36.09	50.02

Tableau 5.2. Valeur en risque historique

Code GAUSS 5.1 – *Calcul de la valeur en risque historique* –

```
/*
**> historicalVaR
**
*/

proc (1) = historicalVaR(data,P,theta,alpha);
  local r,thetaStar,VaR,i;

  data = ln(data);
  r = packr(data - lag1(data));
  thetaStar = theta .* P;
  VaR = zeros(rows(alpha),cols(thetaStar));
  i = 1;
  do until i > cols(VaR);
    VaR[.,i] = - _rndCopulaEmpiricalQuantile(r*thetaStar[.,i],1-alpha);
    i = i + 1;
  endo;

  retp(VaR);
endp;
```

5.2.2 VaR gaussienne

Dans le cas de la VaR gaussienne, nous supposons que $\mathbf{r}(t+1) \sim \mathcal{N}(\mu, \Sigma)$. Nous en déduisons que

$$\Pi(t+1 \mid t) \sim \mathcal{N}\left(\theta_\star(t)^\top \mu, \theta_\star(t)^\top \Sigma \theta_\star(t)\right)$$

Nous obtenons finalement

$$\Phi\left(\frac{x - \theta_\star(t)^\top \mu}{\sqrt{\theta_\star(t)^\top \Sigma \theta_\star(t)}}\right) = 1-\alpha$$

et donc

$$Var\left(\alpha\right) = -\theta_{\star}\left(t\right)^{\top}\mu + \Phi^{-1}\left(\alpha\right)\sqrt{\theta_{\star}\left(t\right)^{\top}\Sigma\theta_{\star}\left(t\right)}.$$

	90%	95%	99%	99.5%	99.9%
P_1	7.185	9.192	12.96	14.33	17.18
P_2	4.082	5.241	7.417	8.213	9.855
P_3	14.18	18.19	25.70	28.45	34.12

Tableau 5.3. Valeur en risque gaussienne

Code GAUSS 5.2 – *Calcul de la valeur en risque gaussienne* –

```
/*
**> gaussianVaR
**
*/

proc (1) = gaussianVaR(data,P,theta,alpha);
  local r,mu,sigma,thetaStar,VaR,i;

  data = ln(data);
  r = packr(data - lag1(data));
  mu = meanc(r);
  Sigma = vcx(r);

  thetaStar = theta .* P;
  VaR = zeros(rows(alpha),cols(thetaStar));
  i = 1;
  do until i > cols(VaR);
    VaR[.,i] = - thetaStar[.,i]'mu +
               cdfni(alpha) * sqrt(thetaStar[.,i]'Sigma*thetaStar[.,i]);
    i = i + 1;
  endo;

  retp(VaR);
endp;
```

5.2.3 VaR paramétrique

La VaR paramétrique est évaluée par la méthode de Monte Carlo. Son calcul repose sur les étapes suivantes :

1. Estimer les paramètres μ_1,...,μ_n des lois marginales $F_1(.;\mu_1)$,...,$F_n(.;\mu_n)$ et le paramètre μ_C de la copule $\mathbf{C}(.;\mu_C)$.
2. Réaliser Ns simulations de vecteurs $(u_1^j, ..., u_n^j), j \in \{1, ..., Ns\}$ suivant la copule $\mathbf{C}(.;\hat{\mu}_C)$:
   - A chaque étape de la simulation, calculer $r_i^j(t+1) = F_i^{-1}(u_i^j;\hat{\mu}_i)$, $i = 1, ..., n$;
   - On en déduit $\Pi^j\left(t+1 \mid t\right) = \theta_{\star}\left(t\right)^{\top}\mathbf{r}^j\left(t+1\right)$.
3. Calculer $Var(\alpha)$ par la méthode des quantiles empiriques à partir de la distribution estimée

$$\Pi^1\left(t+1 \mid t\right), ..., \Pi^j\left(t+1 \mid t\right), ..., \Pi^{Ns}\left(t+1 \mid t\right).$$

Nous reportons les valeurs en risque obtenues pour différentes spécifications dans les tables 5.4, 5.5 et 5.6. Par rapport à la valeur en risque analytique (ou gaussienne), une valeur en risque construite à partir de distributions et d'une copule plus risquées[1] n'est pas systématiquement plus élevée. Dans notre exemple, pour α égal à 90%, la VaR gaussienne[2] est plus élevée que les deux autres. Ceci est vrai aussi pour α égal à 95% dans de nombreux cas. C'est à partir de 99% que la VaR gaussienne commence à être moins élevée que les deux autres VaRs, et pour des quantiles extrêmes, la différence peut être importante.

	90%	95%	99%	99.5%	99.9%
P_1	7.340	9.402	13.20	14.65	17.54
P_2	4.047	5.176	7.318	8.107	9.753
P_3	13.96	17.90	25.21	27.92	33.54

Tableau 5.4. Marges gaussiennes et copule Normale

	90%	95%	99%	99.5%	99.9%
P_1	5.797	8.112	13.36	15.65	20.15
P_2	3.812	5.531	9.801	11.65	16.34
P_3	13.45	19.32	34.15	40.77	54.95

Tableau 5.5. Marges gaussiennes et copule Student $\mathbf{t}_1$

	90%	95%	99%	99.5%	99.9%
P_1	6.593	8.885	14.33	16.98	23.96
P_2	3.778	5.021	7.928	9.351	13.42
P_3	12.80	17.13	27.52	32.89	49.09

Tableau 5.6. Marges student $\mathbf{t}_4$ et copule Normale

Voici le programme qui calcule les valeurs en risque de l'exemple LME.

Code GAUSS 5.3 – *Calcul de la valeur en risque paramétrique* –

```
new;
library copula,pgraph,optmum;
CopulaSet;

dataset = ''lme'';
let vars  = AL AL-15 CU NI PB;
data = LoadData(dataset,vars);
dates = LoadData(dataset,''date'');

let theta[3,5] =  1  1  1 1  1
                 -1 -1 -1 1  1
                  2  1 -3 4 -5;
```

[1] Terme abusif pour dire que les marges sont des distributions à queues épaisses et la copule présente une dépendance de queue – par forcément asymptotique — plus grande que celle de la copule Normale.

[2] Notons que nous obtenons de légères différences entre les tables 5.4 et 5.3 page précédente à cause du Monte Carlo.

```
theta = theta';
let P =100 100 100 100 100;
thetaStar = theta .* P;

let alpha = 0.90 0.95 0.99 0.995 0.999;

VaR = zeros(rows(alpha),cols(thetaStar));

ns = 300000; nu = 1;

data = ln(data);
r = packr(data - lag1(data));
mu = meanc(r);
sigma = stdc(r);

u = dependogram(r);
{rhoStudent,rhoNormal} = regCopulaStudent(substute(u,u.==1,1-__macheps),nu);

format 8,4;

output file = chap8-4.out reset;

/*
**> Normal copula + Gaussian distributions
**
*/

rndseed 123;

u = rndCopulaNormal(rhoNormal,ns);
r = mu' + cdfni(u) .* sigma';
i = 1;
do until i > cols(VaR);
  VaR[.,i] = - _rndCopulaEmpiricalQuantile(r*thetaStar[.,i],1-alpha);
  i = i + 1;
endo;

print VaR';
clear u,r;

/*
**> Student copula + Gaussian distributions
**
*/

rndseed 123;

u = rndCopulaStudent(rhoStudent,nu,ns);
r = mu' + cdfni(u).*sigma';
i = 1;
do until i > cols(VaR);
  VaR[.,i] = - _rndCopulaEmpiricalQuantile(r*thetaStar[.,i],1-alpha);
```

```
  i = i + 1;
endo;

print VaR';
clear u,r;

/*
**> Normal copula + Student distributions
**
*/

rndseed 123;

df = 4;
sigma = sigma/sqrt(df/(df-2));

u = rndCopulaNormal(rhoNormal,ns);
r = mu' + cdfti(u,df).*sigma';
i = 1;
do until i > cols(VaR);
  VaR[.,i] = - _rndCopulaEmpiricalQuantile(r*thetaStar[.,i],1-alpha);
  i = i + 1;
endo;

print VaR';
clear u,r;

output off;
```

5.2.4 VaR semi-historique

Nous proposons dans cette section une utilisation originale des copules pour calculer la valeur en risque. Il s'agit d'une VaR semi-historique puisque :
- la dépendance des facteurs est une copule paramétrique (Normale par exemple) ;
- les marges des facteurs sont les distributions empiriques.

On a donc une combinaison de la VaR historique (pour les directions unidimensionnelles) et de la VaR paramétrique (pour la dépendance des directions unidimensionnelles).

Remarque 5.1 *Cette VaR peut être intéressante lorsque les facteurs de risque ne sont pas évalués sur les mêmes périodes de temps. Dans ce cas, Il n'est pas possible de construire une VaR historique. On peut construire une VaR gaussienne en prenant des corrélations conservatrices (à dire d'expert par exemple). Il est à craindre que celle-ci sous-estime fortement le risque puisque les marges sont gaussiennes. La VaR semi-historique est donc une VaR historique avec un mécanisme de dépendance paramétrique ou une VaR paramétrique (au sens où la copule est paramétrique, par exemple Normale) avec des marges historiques (c'est-à-dire empiriques).*

Pour la VaR semi-historique, la distribution des rendements est construite avec les marges empiriques $\hat{F}_1$,...,$\hat{F}_n$ et une copule paramétrique. $\mathbf{F}^{-1}(1-\alpha)$ est ensuite calculé par la méthode de Monte Carlo. Plus précisément, le calcul de la VaR semi-historique repose sur les étapes suivantes :

1. Estimer le μ_C de la copule $\mathbf{C}(.;\mu_C)$.
2. Réaliser Ns simulations de vecteurs $(u_1^j, ..., u_n^j), j \in \{1, ..., Ns\}$ suivant la copule $\mathbf{C}(.;\hat{\mu}_C)$:

- A chaque étape de la simulation, calculer $r_i^j(t+1) = \hat{F}_i^{-1}(u_i^j)$ par la méthode des quantiles empiriques, $i = 1, ..., n$;
- On en déduit $\Pi^j\left(t+1 \mid t\right) = \theta_\star\left(t\right)^\top \mathbf{r}^j\left(t+1\right)$.

3. Calculer $Var(\alpha)$ par la méthode des quantiles empiriques à partir de la distribution estimée

$$\Pi^1\left(t+1 \mid t\right), ..., \Pi^j\left(t+1 \mid t\right), ..., \Pi^{Ns}\left(t+1 \mid t\right).$$

Nous avons reporté dans les tables 5.7 et 5.8 les VaRs semi-historiques obtenues. C'est la VaR semi-paramétrique avec une copule Normale qui est la plus proche de la VaR historique.

	90%	95%	99%	99.5%	99.9%
P_1	6.601	8.979	14.98	17.82	24.52
P_2	3.807	5.088	8.140	9.617	13.84
P_3	12.83	17.28	28.28	33.78	46.12

Tableau 5.7. Valeur en risque semi-historique (copule Normale)

	90%	95%	99%	99.5%	99.9%
P_1	5.593	8.142	16.00	20.42	33.63
P_2	2.935	4.460	9.343	12.29	20.69
P_3	10.21	15.48	32.11	41.74	68.60

Tableau 5.8. Valeur en risque semi-historique (copule $\mathbf{t}_1$)

Code GAUSS 5.4 – *Calcul de la valeur en risque semi-historique (copule Normale)* –

```
/*
**> copulaNormalVaR
**
*/

proc (1) = copulaNormalVaR(data,P,theta,alpha,ns);
  local r,u,rho,thetaStar,VaR,i;

  data = ln(data);
  r = packr(data - lag1(data));
  u = dependogram(r);
  rho = regCopulaNormal(substute(u,u.==1,1-__macheps));
  thetaStar = theta .* P;
  r = rndCopulaEmpiricalQuantile(rndCopulaNormal(rho,ns),r);

  VaR = zeros(rows(alpha),cols(thetaStar));
  i = 1;
  do until i > cols(VaR);
    VaR[.,i] = - _rndCopulaEmpiricalQuantile(r*thetaStar[.,i],1-alpha);
    i = i + 1;
  endo;

  retp(VaR);
endp;
```

Code GAUSS 5.5 – *Calcul de la valeur en risque semi-historique (copule Student)* –

```
/*
**> copulaStudentVaR
**
*/

proc (1) = copulaStudentVaR(data,P,theta,alpha,nu,ns);
  local r,u,rhoStudent,rho,thetaStar,VaR,i;

  data = ln(data);
  r = packr(data - lag1(data));
  u = dependogram(r);
  {rhoStudent,rho} = regCopulaStudent(substute(u,u.==1,1-__macheps),nu);
  thetaStar = theta .* P;
  r = rndCopulaEmpiricalQuantile(rndCopulaStudent(rho,nu,ns),r);

  VaR = zeros(rows(alpha),cols(thetaStar));
  i = 1;
  do until i > cols(VaR);
    VaR[.,i] = - _rndCopulaEmpiricalQuantile(r*thetaStar[.,i],1-alpha);
    i = i + 1;
  endo;

  retp(VaR);
endp;
```

5.3 Calcul de la charge en capital pour le risque opérationnel

La méthode LDA (voir [3, Frachot, Georges et Roncalli,2001]) cherche à modéliser la perte totale des risques opérationnels (pour un type de risque donné et une ligne de métier donnée) qui est perçue comme une perte agrégée (**aggregate loss distribution**) de différents événements. Celle-ci se définit donc par :

1. le nombre de pertes individuelles,
2. et le montant de chaque pertes individuelles.

A priori, nous ne connaissons pas le nombre de pertes ainsi que le montant des pertes pour la future année. C'est pourquoi nous allons les considérer aléatoires. Nous allons donc modéliser ce nombre de pertes N par un processus de comptage : la distribution $\mathbf{P}$ de ce nombre de pertes est appelée la distribution de la fréquence des pertes (**loss frequency distribution**). Nous supposons aussi que les pertes individuelles sont indépendantes et identiquement distribuées. La distribution du montant d'une perte individuelle ζ est appelée la distribution de la sévérité des pertes $\mathbf{F}$ (**loss severity distribution**). Dans ce cas là, la perte agrégée ϑ est la somme **aléatoire** (car N est aléatoire) des pertes individuelles :

$$\vartheta = \sum_{n=1}^{N} \zeta_n.$$

La distribution $\mathbf{G}$ de la perte agrégée est donc une distribution composée (**compound distribution**) :

$$\mathbf{G}(x) = \mathbb{P}\{N=1\} \times \mathbf{F}(x) + \mathbb{P}\{N=2\} \times \mathbf{F}^{2\star}(x) + \mathbb{P}\{N=3\} \times \mathbf{F}^{3\star}(x) + \ldots.$$

$\mathbf{F}^{n\star}$ est la distribution de la somme de n pertes $\zeta_1, \ldots, \zeta_n$. La charge en capital réglementaire (Capital-at-Risk) correspond à la valeur en risque au seuil de confiance α :

$$CaR(\alpha) = \mathbf{G}^{-1}(\alpha)$$

Pour information, le Comité de Bâle a fixé α égal à 99.9% dans son dernier document consultatif.

Remarque 5.2 *Même si cela n'est pas l'objet du cours, précisons ce que peut être un risque de sévérité. Pour cela, nous utilisons les données réelles du Crédit Lyonnais. Pour un type de risque, nous calculons le rapport entre la perte maximale observée et le quantile α empirique. Nous obtenons les résultats suivants :*

α	*Maximum/Quantile*
50%	$\simeq 160000$
75%	$\simeq 27000$
90%	$\simeq 13825$

Bien sûr, ce type de risque présente des rapports Maximum/Quantile peu communs. Néanmoins, cela illustre parfaitement la nature de nombreux types de risque opérationnel. A titre de comparaison, le rapport du quantile 99.9999% de la distribution gaussienne sur le quantile 90% de la distribution gaussienne est beaucoup plus faible :

$$\frac{\Phi^{-1}(99.9999\%)}{\Phi^{-1}(90\%)} = 3.7$$

En fait, le risque opérationnel est sûrement le risque qui présente le plus d'événements rares. Le choix de la distribution de sévérité est donc crucial. Généralement, les distributions les plus utilisées sont les lois lognormale, Pareto, Weibull, GEV.

L'agrégation est un problème relativement difficile dans le cas du risque opérationnel. Bien sûr, nous pouvons faire l'hypothèse que les différents types de risque sont indépendants. Dans ce cas, le problème est résolu. Si nous voulons prendre en compte la dépendance entre les risques, nous pouvons

- soit corréler les fréquences ;
- soit corréler les pertes agrégées.

Corrélation des fréquences

La première méthode a été proposée par [2, Bouyé et al., 2000]. L'idée est de construire des distributions multidimensionnelles de variables aléatoires de Poisson à partir des copules :

$$\begin{aligned}
\mathbb{P}\{N_1 = n_1, N_2 = n_2\} = \quad & \mathbf{C}\left(\sum_{n=0}^{n_1} \frac{\lambda_1^n e^{-\lambda_1}}{n!}, \sum_{n=0}^{n_2} \frac{\lambda_2^n e^{-\lambda_2}}{n!}\right) \\
- \quad & \mathbf{C}\left(\sum_{n=0}^{n_1-1} \frac{\lambda_1^n e^{-\lambda_1}}{n!}, \sum_{n=0}^{n_2} \frac{\lambda_2^n e^{-\lambda_2}}{n!}\right) \\
- \quad & \mathbf{C}\left(\sum_{n=0}^{n_1} \frac{\lambda_1^n e^{-\lambda_1}}{n!}, \sum_{n=0}^{n_2-1} \frac{\lambda_2^n e^{-\lambda_2}}{n!}\right) \\
+ \quad & \mathbf{C}\left(\sum_{n=0}^{n_1-1} \frac{\lambda_1^n e^{-\lambda_1}}{n!}, \sum_{n=0}^{n_2-1} \frac{\lambda_2^n e^{-\lambda_2}}{n!}\right)
\end{aligned}$$

A titre d'illustration, les tables 5.9 et 5.10 correspondent aux probabilités $p_{i,j} = \mathbb{P}\{N_1 = i, N_2 = j\}$ de la distribution de Poisson bivariée avec $\lambda_1 = 1$, $\lambda_2 = 1$ et une copule Normale de paramètre ρ.

Néanmoins, les simulations que nous avons faites montrent que le paramètre ρ de la copule Normale a peu d'influence sur la mesure de risque. Les explications sont les suivantes :

- Le choix de la copule Normale ne permet pas de mettre de la dépendance sur les extrêmes. Une copule de valeurs extrêmes a une influence plus importante sur la mesure de risque.
- L'absence de dépendance sur les extrêmes est accentuée par deux phénomènes : les fréquences sont discrètes (ce qui veut dire par exemple que la mesure de risque sera plus sensible à la copule si les fréquences de Poisson sont importantes, car le caractère discret de la distribution est moins accentué) et la mesure de risque porte sur la distribution agrégée. Nous pouvons très bien avoir deux extrêmes N_1 et N_2, cela ne veut pas dire que nous obtenons nécessairement un extrême pour la somme des deux distributions composées, car le risque opérationnel est avant tout **un risque de sévérité** (quelques pertes extrêmes suffisent à représenter 95% du risque opérationnel).

$p_{i,j}$	0	1	2	3	4	5	$\cdots$	$p_{i,\cdot}$
0	0.0945	0.133	0.0885	0.0376	0.0114	0.00268		0.368
1	0.0336	0.1	0.113	0.0739	0.0326	0.0107		0.368
2	0.00637	0.0312	0.0523	0.0478	0.0286	0.0123		0.184
3	0.000795	0.00585	0.0137	0.0167	0.013	0.0071		0.0613
4	7.28E-005	0.000767	0.00241	0.00381	0.00373	0.00254		0.0153
5	5.21E-006	7.6E-005	0.000312	0.000625	0.000759	0.000629		0.00307
$\vdots$								
$p_{\cdot,j}$	0.135	0.271	0.271	0.18	0.0902	0.0361		1

Tableau 5.9. Loi de Poisson bidimensionnelle avec $\rho = 0.5$

$p_{i,j}$	0	1	2	3	4	5	$\cdots$	$p_{i,\cdot}$
0	0.0136	0.0617	0.101	0.0929	0.058	0.027		0.368
1	0.0439	0.112	0.111	0.0649	0.026	0.00775		0.368
2	0.0441	0.0683	0.0458	0.0188	0.00548	0.00121		0.184
3	0.0234	0.0229	0.0109	0.00331	0.000733	0.000126		0.0613
4	0.00804	0.00505	0.00175	0.000407	7.06E-005	9.71E-006		0.0153
5	0.002	0.00081	0.000209	3.79E-005	5.26E-006	5.89E-007		0.00307
$\vdots$								
$p_{\cdot,j}$	0.135	0.271	0.271	0.18	0.0902	0.0361		1

Tableau 5.10. Loi de Poisson bidimensionnelle avec $\rho = -0.5$

Corrélation des pertes

Dans [3, Frachot, Georges et Roncalli , 2001], les auteurs présentent une seconde méthode pour agréger les risques qui est beaucoup plus acceptable. La dépendance est directement mise sur les pertes. Soient ϑ_1 et ϑ_2 deux pertes correspondant à deux types différents de risque opérationnel. Soit $\mathbf{G}$ la distribution jointe. Nous avons

$$\mathbf{G}(x_1, x_2) = \mathbf{C}(\mathbf{G}_1(x_1), \mathbf{G}_2(x_2))$$

Dans ce cas, $\mathbf{G}_1$ et $\mathbf{G}_2$ sont les deux distributions composées qui correspondent aux variables aléatoires ϑ_1 et ϑ_2. Soit $\mathbf{G}_{1+2}$ la distribution de $\vartheta_1 + \vartheta_2$. Nous avons

$$\begin{aligned} \mathbf{G}_{1+2}(x) &= \mathbb{P}\{\vartheta_1 + \vartheta_2 \leq x\} \\ &= \iint_{x_1+x_2 \leq x} \mathrm{d}\mathbf{C}(\mathbf{G}_1(x_1), \mathbf{G}_2(x_2)) \end{aligned}$$

et

$$CaR(\alpha) = \mathbf{G}_{1+2}^{-1}(\alpha).$$

La grande difficulté de cette méthode est l'aspect numérique pour obtenir $CaR(\alpha)$. En général, nous n'avons pas d'expression analytique des marges $\mathbf{G}_1$ et $\mathbf{G}_2$. Il est donc difficile de construire la distribution $\mathbf{G}$ et encore plus difficile d'obtenir $\mathbf{G}_{1+2}$ et $CaR(\alpha)$. Néanmoins, nous pouvons contourner ces difficultés en utilisant une méthode de Monte Carlo. Dans ce cas, la simulation de $\mathbf{G}$ se fait par la méthode des quantiles empiriques.

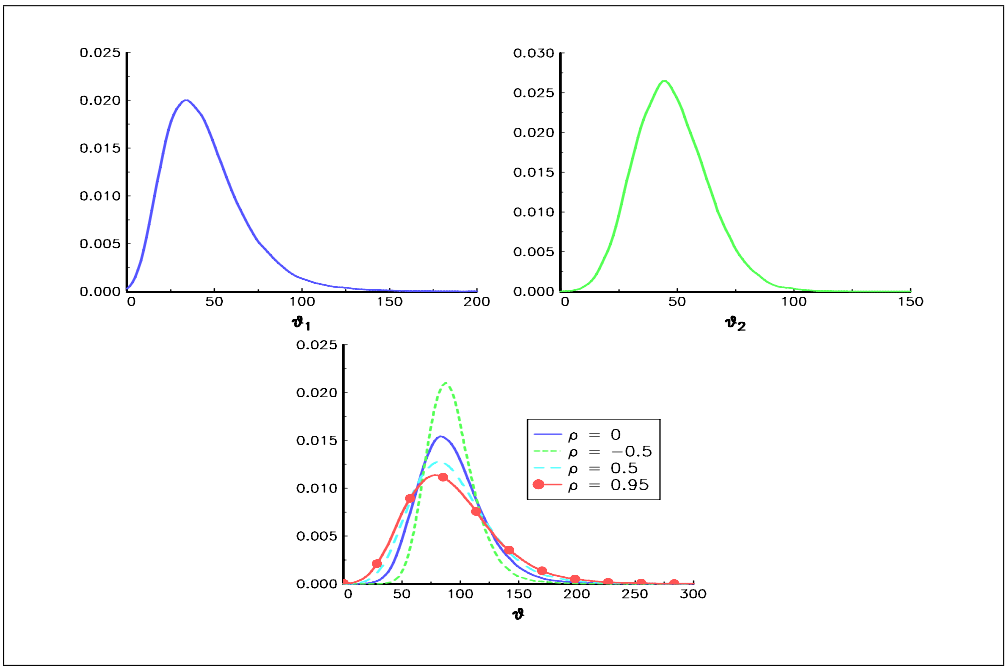
L'exemple suivant est celui donné par [3, textscFrachot, Georges et Roncalli , 2001]. Nous avons

$$\zeta_1 \sim \mathcal{LN}(1,1) \text{ et } \zeta_2 \sim \mathcal{LN}(1.25, 0.5).$$

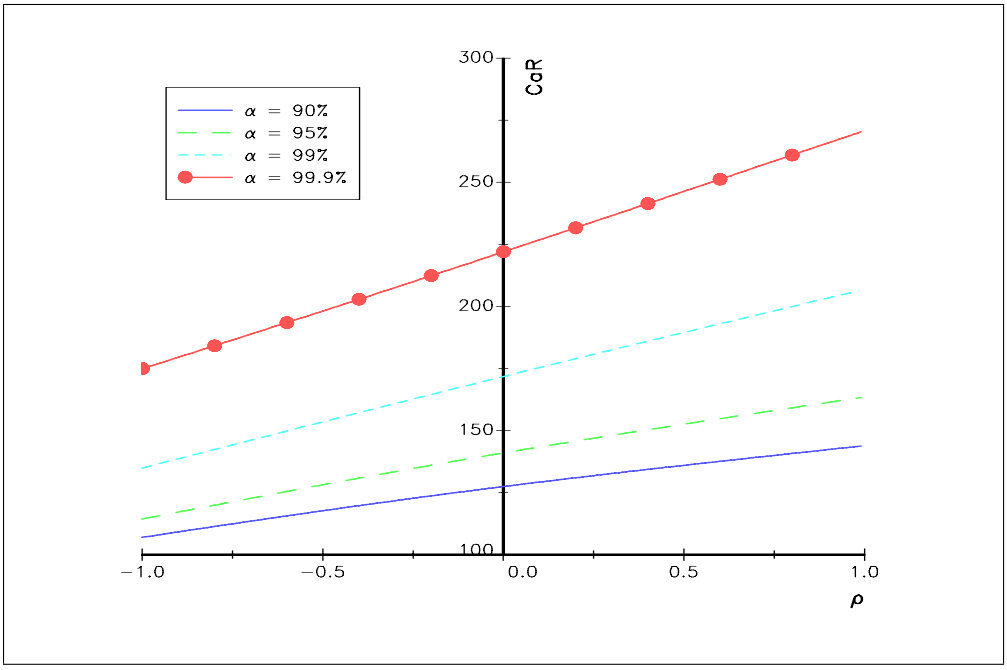
Les distributions de fréquence sont respectivement $\mathcal{P}(10)$ et $\mathcal{P}(12)$.

Sur le graphique 5.4, nous représentons la distribution de la perte totale $\vartheta = \vartheta_1 + \vartheta_2$ pour différentes copules Nomales. La charge en capital correspondante est reportée sur le graphique 5.5.

Il est intéressant de noter que le *Capital-at-Risk* semble être une fonction linéaire du paramètre ρ de la copule. **Encore une fois, cette propriété provient du caractère gaussien de la copule.** De plus, ceci n'est plus forcément vrai si les distributions lognormales ont un paramètre σ plus élevé.



Graphique 5.4. Impact de la fonction de dépendance sur la distribution de la perte totale



Graphique 5.5. Impact du paramètre ρ sur la charge en capital

Code GAUSS 5.6 – *Simulation de ϑ_1 et ϑ_2 avec une copule* Normale *($\zeta_1 \sim \mathcal{LN}(\mu_1, \sigma_1)$, $\zeta_2 \sim \mathcal{LN}(\mu_2, \sigma_2)$, $N_1 \sim \mathcal{P}(\lambda_1)$ et $N_2 \sim \mathcal{P}(\lambda_2)$)* –

```
/*
**> rndCopulaRiskOp
**
*/

proc (1) = rndCopulaRiskOp(mu1,sigma1,lambda1,mu2,sigma2,lambda2,rho,ns);
  local vartheta1,vartheta2,N1,N2,i,u1,u2,vartheta;

  vartheta1 = zeros(ns,1);
  vartheta2 = zeros(ns,1);

  N1 = rndp(ns,1,lambda1);
  N2 = rndp(ns,1,lambda2);

  i = 1;
  do until i > ns;
    if N1[i] /= 0;
      vartheta1[i] = sumc(_rndLN(mu1,sigma1,N1[i],1));
    endif;
    if N2[i] /= 0;
      vartheta2[i] = sumc(_rndLN(mu2,sigma2,N2[i],1));
    endif;
    i = i + 1;
  endo;
```

```
  {u1,u2} = rndCopulaNormal2(rho,ns);
  vartheta = rndCopulaEmpiricalQuantile(u1~u2,vartheta1~vartheta2);

  retp(vartheta);
endp;

proc (1) = _rndLN(mu,sigma,r,c);
  local u;

  u = exp(mu + sigma * rndn(r,c));

  retp(u);
endp;
```

Bibliographie

[1] Bouyé, E., V. Durrleman, A. Nikeghbali, G. Riboulet et T. Roncalli [2000], Copulas for finance — a reading guide and some applications, Groupe de Recherche Opérationnelle, Crédit Lyonnais, *Working Paper*

[2] Bouyé, E., V. Durrleman, A. Nikeghbali, G. Riboulet et T. Roncalli [2000], Copulas : an open field for risk management, Groupe de Recherche Opérationnelle, Crédit Lyonnais, *Working Paper*

[3] Frachot, A., P. Georges et T. Roncalli [2001], Loss Distribution Approach for operational risk, Groupe de Recherche Opérationnelle, Crédit Lyonnais, *Working Paper*

[5] Deheuvels, P. [1978], Caractérisation complète des lois extrêmes multivariées et de la convergence des types extrêmes, *Publications de l'Institut de Statistique de l'Université de Paris*, **23(3-4)**, 1-36

[5] Deheuvels, P. [1981], An asymptotic decomposition for multivariate distribution-free tests of independence, *Journal of Multivariate Analysis*, **11**, 102-113

[6] Devroye, L. [1986], Non-Uniform Random Variate Generation, Springer-Verlag, New York

[7] Devroye, L, Random variate generation in one line of code, *working paper*

[8] Fishman, G.S,, Monte Carlo, Concepts, Algorithms and Applications, Springer-Verlag

[9] Genest, C. [1987], Frank's family of bivariate distributions, *Biometrika*, **74**, 549-555

[10] Genest, C. et J. MacKay [1986], Copules archimédiennes et familles de lois bidimensionnelles dont les marges sont données, *Canadian Journal of Statistics*, **14(2)**, 145-159

[11] Genest, C. et J. MacKay [1986], The joy of copulas : Bivariate distributions with uniform marginals, *American Statistician*, **40**, 280-283

[12] Geweke, J. [1988], Antithetic acceleration of Monte Carlo integration in Bayesian inference, *Journal of Econometrics*, **38**, 73–90

[13] Glasserman, P., P. Heidelberger, and P. Shahabuddin [1998], Asymptotically Optimal importance Sampling and Stratification for Pricing Path-Dependent Options

[14] Heston, S.L. [1993], A closed-form solution for options with stochastic volatility with application to bond and currency options, *The Review of Financial Studies*

[15] Joe, H. [1997], Multivariate Models and Dependence Concepts, *Monographs on Statistics and Applied Probability*, **73**, Chapmann & Hall, London

[16] Johnson, M.E. [1987], Multivariate Statistical Simulation, John Wiley & Sons, New York

[17] Kloeden, P.E. and E. Platen [1995] Numerical solution of stochastic differential equations, Strpinger-Verlag

[18] Newton, N.J. [1994] Variance Reduction for Simulated diffusions", *SIAM J. Appl. Math.* **54**, 1780-1805

[19] Nelsen, R.B. [1999], An Introduction to Copulas, *Lectures Notes in Statistics*, **139**, Springer-Verlag, New York

[20] Ostrowski, A.M. [1973], Solutions to equations in Euclidean and Banach spaces, *Academic Press*, New York

[21] Casella, G, and C.P. Robert [2000], Monte Carlo Statistical Methods, Springer-Verlag

[22] Rubinstein, R.Y., [1981] Simulation and the Monte Carlo Method, J. Wiley, New York.

[23] Press, W.H., S.A. Teukolsky, W.T. Vetterling et B.P. Flannery [1992], Numerical Recipes in Fortran, second edition, *Cambridge University Press*, Cambridge

[24] Shorack, G.R. et J.A. Wellner [1986], *Empirical Processes with Applications to Statistics*, John Wiley & Sons, New York