

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaïa



Faculté des Sciences Exactes
Département Mathématique/MI

Mémoire de fin de cycle

En vue de l'obtention du diplôme de
Master en Mathématiques

Option

Statistique et Analyse décisionnelle

Thème

Les méthodes Monte Carlo, Quasi Monte Carlo et leurs applications

Réalisé par :

M^{elle} DERRADJI NAWEL

M^{elle} KHALFI LINDA.

Devant le jury composé de :

Président	M ^r ALOUI.A	MCB	U. A/Mira Béjaïa.
Rapporteur	M ^{me} OURBIH.M	Professeur	U. A/Mira Béjaïa.
Examineur	M ^{me} GUEBLIS	MAA	U. A/Mira Béjaïa.

Béjaïa 2013

Introduction générale	7
I Généralités sur les méthodes de Monte Carlo	10
I.1 Introduction	11
I.2 Notions générales	11
I.2.1 Variables aléatoires	11
I.2.2 La loi forte des grands nombres	12
I.2.3 Le théorème central limite	12
I.2.4 Définition de la simulation	13
I.2.5 Simulation des nombres aléatoires	14
I.2.6 Génération des nombres pseudo-aléatoires	14
I.2.7 Tests de l'uniformité de la distribution	15
I.2.8 Tests d'indépendance	17
I.2.9 Qualités d'un bon générateur	18
I.2.10 Quelques générateurs de nombres pseudo-aléatoires	18
I.3 Simulation de variables aléatoires suivant une loi donnée	20
I.3.1 Méthode d'inversion	20
I.3.2 Méthode de rejet	21
I.3.3 Simulation de variable Gaussienne	21
I.4 Simulation de Monte Carlo	22
I.4.1 Description de la méthode	23
I.4.2 Simulation de Monte Carlo : calcul d'intégrale	23
I.4.3 Erreurs dus à la méthode de MC	26

I.4.4	La méthode replicated runs (histoires répliquées)	26
I.4.5	Réduction de la variance	26
I.5	Les méthodes non aléatoires	28
I.5.1	L'échantillonnage Descriptif(ED)	28
I.5.2	L'échantillonnage descriptif amélioré(RDS)	29
I.5.3	L'échantillonnage par Hypercube Latin	31
I.5.4	L'échantillonnage par Supercube Latin	32
I.5.5	Quasi Monte Carlo	32
I.5.6	Quasi Monte Carlo randomisées	33
I.6	Conclusion	33
II	Les méthodes de Quasi Monte Carlo	34
II.1	Introduction	35
II.2	Définition de la méthode Quasi Monte Carlo	35
II.3	Quelques notions utilisées sur les méthodes QMC	36
II.3.1	La discrépance	36
II.3.2	La variation de Hardy Krause	36
II.3.3	Inégalité de Koksma-Hlawka	37
II.3.4	Suites à faible discrépance	37
II.3.5	La fonction radical inverse	37
II.3.6	Les nombres de directions	38
II.3.7	Le code Gray	38
II.3.8	Le polynôme primitif	38
II.3.9	La matrice de Pascal	39
II.4	Etat de l'art	39
II.5	Les suites de QMC	40
II.5.1	Les suites de Van Der Corput	40
II.5.2	Les suites de Halton	41
II.5.3	Les suites de Hammersley	45
II.5.4	Les suites de Sobol	46
II.5.5	Les suites de Faure	50
II.5.6	Good lattice points	53
II.5.7	Quasi Monte Carlo randomisées	54
II.6	Conclusion	55
III	Application et comparaison	57
III.1	Introduction	58

III.2	Présentation du logiciel MATLAB	58
III.3	Génération des suites de QMC	58
III.3.1	La suite de Halton	58
III.3.2	La suite de Sobol	60
III.3.3	La suite de Faure	61
III.4	Génération d'échantillons exponentielle	62
III.4.1	La suite de Halton	62
III.4.2	La suite de Sobol	63
III.4.3	La suite de Faure	64
III.5	Comparaison entre les méthodes QMC, MC et RDS	65
	Conclusion générale	68
	Bibliographie	74
	Annexes	75

Table des figures

I.1	Schéma du modèle de simulation.	13
I.2	Organigramme des différentes étapes de simulation.	14
I.3	Représentation graphique d'un ensemble de points d'un échantillonnage par LHS	32
II.1	Représentation graphique des 10 premiers nombres de la suite de Halton .	43
II.2	Les 1000 nombres de la suite de Halton en dimension $14 * 15$	43
II.3	Les 1000 nombres de la suite de Halton en dimension $27 * 28$	44
II.4	Les 500 nombres de la suite de Hammersley.	46
II.5	Les 1000 nombres de la suite de Sobol en dimension $1 * 2$	49
II.6	Les 1000 nombres de la suite de Sobol en dimension $39 * 40$	49
II.7	Les 1000 nombres de la suite de Faure en dimension $1 * 2$	52
II.8	Les 1000 nombres de la suite de Faure en dimension $13 * 14$	53
II.9	Les 21 premiers termes d'un good lattice points en dimension 2.	54
III.1	Table de Kolmogorov-Smirnov	76
III.2	Table de Khi-deux	76

Liste des tableaux

I.1	Résultats de l'exemple de la méthode ED	29
I.2	Résultats de l'exemple de la méthode RDS	30
II.1	Les 10 premiers nombres de la suite de Halton	42
II.2	La suite de Sobol dans les 3 premières dimensions.	48
II.3	La suite de Faure dans les 3 premières dimensions.	52
III.1	La suite de Halton en base 2, 7 et 17.	59
III.2	Les résultats du test de Kolmogorov-Smirnov d'uniformité pour $n = 10$. . .	59
III.3	Les résultats du test de Kolmogorov-Smirnov d'uniformité pour $n = 1000$. .	59
III.4	La suite de Sobol en dimension 5, 20 et 190.	60
III.5	Les résultats du test d'uniformité pour $n = 10$ de la suite de Sobol.	60
III.6	Les résultats du test d'uniformité pour $n = 1000$ pour la suite de Sobol. . .	61
III.7	La suite de Faure en dimension 3, 7 et 23.	61
III.8	Les résultats du test d'uniformité pour $n = 10$ pour la suite de Faure . . .	61
III.9	Les résultats du test d'uniformité pour $n = 1000$ pour la suite de Faure . .	62
III.10	L'échantillon de la loi exponentielle avec la suite de Halton	62
III.11	Les résultats du test d'adéquation à la loi exponentielle pour $n = 10$	63
III.12	Les résultats du test d'adéquation à la loi exponentielle pour $n = 1000$. . .	63
III.13	Récapitulatif des décisions d'adéquation à la loi exponentielle.	63
III.14	L'échantillon de la loi exponentielle avec la suite de Sobol.	64
III.15	Les résultats du test d'adéquation à la loi exponentielle pour $n = 10$	64
III.16	Les résultats du test d'adéquation à la loi exponentielle pour $n = 1000$. . .	64
III.17	L'échantillon de la loi exponentielle avec la suite de Faure	65

III.18 Les résultats du test d'adéquation à la loi exponentielle pour $n = 10$	65
III.19 Les résultats du test d'adéquation à la loi exponentielle pour $n = 1000$	65
III.20 Calculs de la moyenne empirique, erreur absolue et (MSE) des différents échantillons générés.	66

Introduction générale

La résolution de nombreux problèmes scientifiques nécessite de calculer des sommes, des intégrales, ou encore de résoudre des équations ou des problèmes d'optimisation. Les techniques de calcul direct, encore appelées techniques analytiques, sont très vite dépassées par la complexité des modèles, elles nécessitent souvent des hypothèses trop fortes, de sorte qu'on ne puisse pas les appliquer, ou alors comme dans le cas de calcul de sommes, le nombre d'opérations requis peut être trop important pour être réalisé en un temps raisonnable. On doit alors nécessairement faire appel à des méthodes d'approximation. Cependant, celles-ci requièrent également des hypothèses fortes, bien que moins fortes que pour les méthodes analytiques. De plus, ces méthodes s'avèrent rapidement inefficaces dès que la dimension de l'espace du problème augmente.

Les méthodes de simulation de Monte Carlo peuvent être vues comme des méthodes d'approximation au sens statistique du terme. Comme nous le verrons, ces méthodes sont moins exigeantes en termes d'hypothèses sur le modèle[54].

Du point de vue des applications, les méthodes de Monte Carlo sont aujourd'hui indispensables dans plusieurs domaines. Ces méthodes peuvent en revanche être gourmandes en temps de calcul si elles ne sont pas correctement utilisées.

De plus, utiliser les méthodes de Monte Carlo nécessite un paramètre d'entrée de nature aléatoire, afin d'obtenir un résultat de nature déterministe, via ce qu'on appelle un générateur de nombres pseudo-aléatoires. Ici aussi, une utilisation inappropriée peut induire à des erreurs parfois subtiles.

La méthode de Monte Carlo est une méthode d'intégration numérique fréquemment employée, cependant sa vitesse de convergence est en $o(\frac{1}{\sqrt{n}})$, n étant le nombre de points de la suite utilisée. L'un des enjeux est donc d'obtenir un intervalle de confiance avec la plus grande précision possible pour un temps de simulation donné, par conséquent, pour obtenir une bonne précision, il est nécessaire d'effectuer un grand nombre de simulations[55].

L'objectif ici est de créer des suites qui convergent plus rapidement que les suites de Monte Carlo, les méthodes que nous avons traité ici sont les méthodes Quasi Monte Carlo, qui ont comme avantage attrayant d'exiger moins de temps de calcul, et une convergence plus rapide, car elles utilisent des suites à faible discrédance[8].

Dans notre cas, nous nous intéressons à la simulation de Monte carlo qui sera introduit dans le premier chapitre, nous discuterons ensuite dans le deuxième chapitre les méthodes Quasi Monte carlo, où nous définissons les suites à faible discrédance entrantes dans la construction de la méthode de Quasi Monte Carlo. Dans notre cas, l'application sera portée sur la génération d'une variable aléatoire exponentielle utilisant des suite à faible discrédance puis une comparaison entre les trois méthodes Monte carlo, quasi Monte carlo et l'échantillonnage descriptif amélioré (RDS) par les deux critères de comparaison l'erreur

absolue et l'erreur moyenne quadratique. Pour clore ce travail une conclusion générale sera présentée.

CHAPITRE I

Généralités sur les méthodes de Monte Carlo

I.1 Introduction

Lorsque la résolution mathématique d'un problème donné n'est pas possible, on fait appel à des méthodes d'approximation. Cela permet de modéliser des situations très complexes dont la solution analytique s'avère difficile voir impossible. Parmi ces méthodes d'approximation figure la simulation, qui représente un outil très utile, elle permet de valider où d'invalider des hypothèses, d'obtenir des informations quantitatives, où tout simplement d'explorer le comportement d'un modèle lorsque celui-ci est mal connu où mal compris[47].

Dans ce chapitre, nous allons présenter quelques notions concernant la simulation ainsi que les méthodes aléatoires, plus particulièrement, la simulation de Monte Carlo, et non aléatoires nécessaires à la compréhension des méthodes Quasi Monte Carlo.

I.2 Notions générales

I.2.1 Variables aléatoires

Definition I.1

Soit $(\Omega, \mathfrak{F}, P)$ un espace probabilisé ; on appelle variable aléatoire sur $(\Omega, \mathfrak{F}, P)$, à valeurs dans l'ensemble E (fini ou infini dénombrable) toute application X de Ω dans E telle que $(\forall x \in E) \{\omega; X(\omega) = x\} \in \mathfrak{F}$.

Si $E = \mathbb{Z}^$ on dit que X est entière.*

Si $E = \mathbb{N}$ on dit que X est entière positive[51].

Nous distinguons deux types de variables aléatoires : discrètes et continues.

Variables aléatoires discrètes

Une variable aléatoire est dite discrète quand $\Omega = \mathbb{N}$,

$$X : \mathbb{N} \rightarrow \mathbb{R}$$

$$i \mapsto X(i) = x_i \in \mathbb{R}.$$

Variables aléatoires continues

C'est une variable aléatoire réelle ie : $(E, \mathfrak{F}) = (\mathbb{R}, B(\mathbb{R}))$ et telle que $\Omega = \mathbb{R}$ ou \mathbb{R}^d ou un intervalle de ces espaces[12].

I.2.2 La loi forte des grands nombres

Les méthodes de Monte Carlo sont basées sur la loi forte des grands nombres :

Soit Y une variable aléatoire, on simule un grand nombre de variables aléatoires (v.a.) $(Y_n)_n$ indépendantes et de même loi que Y , on prend ensuite la moyenne des valeurs prises, $n^{-1} \sum_{i=1}^n Y_i$.

Soient Y_1, Y_2, \dots, Y_n des n réalisations indépendantes identiquement distribuées (i.i.d) de la variable Y , l'estimateur de Monte Carlo est défini comme la moyenne empirique :

$$\hat{\theta}_n = g\left(n^{-1} \sum_{i=1}^n Y_i\right) = g(\bar{Y}_n) \quad (\text{I.1})$$

Par la loi forte des grands nombres, \bar{Y}_n est une suite d'estimateurs de $\mathbb{E}[Y]$, au sens où $\bar{Y}_n \xrightarrow{p.s.} \mathbb{E}[Y]$. Ceci est démontré dans le théorème suivant :

Théorème I.1

Soit $\{Y_n\}_{n>0}$ une suite de v.a. i.i.d. telle que $\mathbb{E}\|Y_i\| < \infty$, Alors

$$\bar{Y}_n = n^{-1} \sum_{i=1}^n Y_i \xrightarrow{p.s.} \mathbb{E}[Y]. \quad (\text{I.2})$$

Si g est une fonction continue au point θ , le théorème de continuité montre

$$\hat{\theta}_n = g(\bar{Y}_n) \xrightarrow{p.s.} g(\mathbb{E}[Y]) = \theta[24]. \quad (\text{I.3})$$

I.2.3 Le théorème central limite

Le théorème central limite permet de construire des intervalles de confiance asymptotiques.

Théorème I.2

Soit $\{Y_n\}_{n \geq 1}$ une suite de v.a. réelles i.i.d. de carré intégrable. On suppose que $\sigma^2 =$

$$\text{Var}(Y_i) > 0 \text{ et on pose : } \bar{Y}_n = \frac{1}{n} \sum_{i=1}^n Y_i \quad \text{et} \quad \hat{\sigma}_n = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y}_n)^2}.$$

Alors,

$$\sqrt{n} \left(\frac{\bar{Y}_n - \mathbb{E}[Y_i]}{\hat{\sigma}_n} \right) \mathbb{1}_{\hat{\sigma}_n > 0} \xrightarrow[n \rightarrow +\infty]{} \mathcal{N}(0, 1) \quad \text{converge en loi.}$$

En particulier, pour tout réel $c > 0$,

$$\mathbb{P} \left[\sqrt{n} \left| \frac{\bar{Y}_n - \mathbb{E}[Y_i]}{\hat{\sigma}_n} \right| \leq c \right] \xrightarrow[n \rightarrow +\infty]{} 1 - \alpha_c. \quad (\text{I.4})$$

Où $\alpha_c = \mathbb{P}[|X| > c]$ pour $X \sim \mathcal{N}(0, 1)$.

Pour n suffisamment grand, la probabilité d'avoir $\mathbb{E}[Y_i] \in [\bar{Y}_n \pm c\hat{\sigma}_n/\sqrt{n}]$ est proche de $1 - \alpha_c$.

Remarque : Une estimation seule ne veut rien dire. L'estimateur est une variable aléatoire qui a sa propre variance. Deux estimations différentes peuvent conduire à deux valeurs très différentes. Il est donc important de connaître la variance de l'estimateur ou, et c'est encore mieux, de fournir un intervalle de confiance, de manière à pouvoir juger la précision du résultat obtenu[16].

I.2.4 Définition de la simulation

La simulation est une technique de modélisation du monde réel, elle permet de représenter le fonctionnement d'un système composé de différents centres d'activités, de mettre en évidence les caractéristiques de ceux-ci et les interactions entre eux, de décrire la circulation des différents objets traités par ces processus, et enfin d'observer le comportement du système dans son ensemble et son évolution dans le temps.

Le schéma suivant illustre le modèle de simulation :

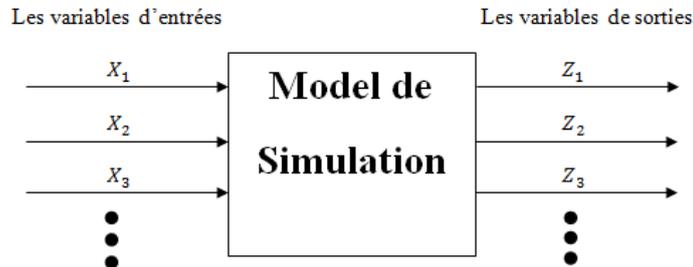


FIGURE I.1 – Schéma du modèle de simulation.

La formalisation du phénomène peut s'effectuer aux moyens de modèles mathéma-

tiques qui présentent l'avantage de fournir des résultats reflétant une évolution stable du système dans le temps. Les étapes de la simulation sont exécutées dans l'ordre suivant : Formulation du problème, modélisation, validation du modèle, programmation, vérification du problème, simulation enfin l'analyse et interprétation des résultats. Ces étapes sont illustrées dans l'organigramme suivant[13] :

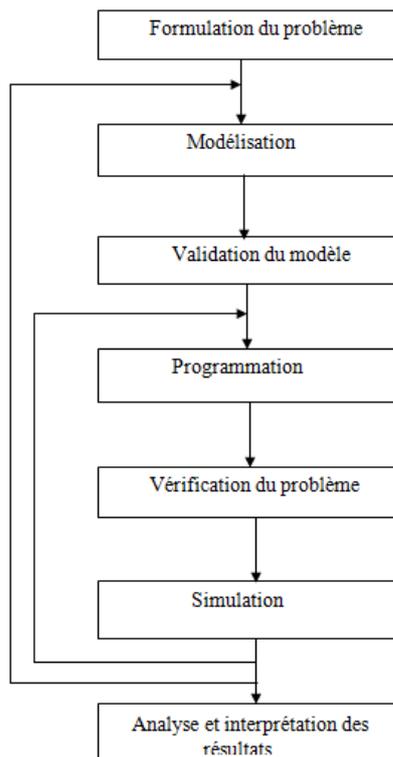


FIGURE I.2 – Organigramme des différentes étapes de simulation.

I.2.5 Simulation des nombres aléatoires

La simulation est une expérience faite dans un environnement de hasard. A l'origine cette expérience reposait sur le tirage des boules dans une urne numérotés de 1 à 9, afin d'obtenir des tables de nombres aléatoires, ceci étant insuffisant, il y a eu l'apparition de machines générant ces nombres aléatoires[17].

I.2.6 Génération des nombres pseudo-aléatoires

Toute simulation fait intervenir des nombres au hasard et il est donc nécessaire de tester le comportement des suites générées selon les deux critères suivant :

1. **Uniformité** : Peut-on considérer les résultats d'une génération comme des réalisations d'une variable aléatoire uniforme sur $[0, 1]$?
2. **Indépendance** : Peut-on considérer les nombres aléatoires générés \mathcal{U}_n comme des réalisations de variable aléatoire indépendante ? [41].

Afin de répondre à ces deux questions on doit passer un certain nombre de tests statistiques d'adéquation à la loi uniforme $\mathcal{U}_{[0,1]}$ et d'indépendance. Rappelons les tests classiques d'adéquation sur la suite (u_1, u_2, \dots, u_n) de variables aléatoires simulées [40].

I.2.7 Tests de l'uniformité de la distribution

L'idée est de comparer une loi empirique (obtenue à partir des nombres générés) avec une loi théorique ($\mathcal{U}_{[0,1]}$).

H_0 : "les observations fournies par le générateur $\rightsquigarrow \mathcal{U}_{[0,1]}$ ".

H_1 : "les observations fournies par le générateur ne suivent pas la loi $\mathcal{U}_{[0,1]}$ ".

Test de khi deux

L'idée est de découper l'espace de valeurs de X en k classes différentes en $[0, 1]$, de probabilité p_1, \dots, p_k .

Soient :

n : le nombre d'observations pseudo-aléatoires générés.

k : le nombre de classes de l'intervalle $[0, 1]$.

n_i : les effectifs observés de la $i^{\text{ème}}$ classe.

p_i : la probabilité associée à la classe.

$np_i = \frac{n}{k}$: les effectifs théoriques de chaque classe (loi uniforme).

On considère la statistique du test :

$$T_n = \frac{k}{n} \sum_{i=1}^k (n_i - \frac{n}{k})^2 = \frac{\sum_{i=1}^k (n_i - np_i)^2}{np_i}. \quad (\text{I.5})$$

T_n suit approximativement, sous l'hypothèse H_0 , la loi de khi-deux à $(k - 1)$ degrés de liberté (ddl) (χ_{k-1}^2).

On fixe un seuil de signification α , et on rejette l'hypothèse H_0 , si $T_n > \chi_{k-1}^2(\alpha)$. (valeur critique lue sur la table de χ_{k-1}^2).

Conditions d'utilisation :

- Pour que ce test ait un sens il faut que $np_i > 5$.
- Si $k > 30$, on utilise l'approximation de Fisher de la loi de χ^2 par la loi normale ie : $n \geq 5k$.

– Pour le calcul de k , on peut utiliser les approximations suivantes :

Approximation de Mann et Wald : $k = (4n)^{\frac{2}{5}}$.

Approximation de Strunges : $k = 1 + 3,3 \ln(n)$.

Exemple :

Soit la suite de nombre aléatoires sur $[0, 1]$.

0,99 – 0,01 – 0,03 – 0,05 – 0,89 – 0,91 – 0,19 – 0,33 – 0,36 – 0,45 – 0,80 – 0,70 –
0,75 – 0,20 – 0,31 – 0,23 – 0,65 – 0,35 – 0,43 – 0,42 – 0,62 – 0,11 – 0,15 – 0,79 – 0,21.

La taille n de l'échantillon est 25, $k = 4$.

Classes	$[0 - 0,25[$	$[0,25 - 0,50[$	$[0,50 - 0,75[$	$[0,75 - 1[$	Total
n_i	9	7	3	6	25
$\frac{n_i}{k}$	6,25	6,25	6,25	6,25	
$(n_i - \frac{n_i}{k})^2$	7,56	0,56	10,56	0,0625	18,75

TABLEAU I.1- Résultats du test de khi-deux

$T_n = 3$; pour un seuil de signification $\alpha = 0,05$, la valeur critique lue sur la table du χ_3^2 est de 7,815 et comme $T_n < \chi_3^2(\alpha)$ on accepte H_0 [41].

Test de kolmogorov-Smirnov

Ce test est basé sur le calcul de la distance entre la fonction de répartition empirique $F_n(x)$ et la fonction de répartition théorique $F(x)$ de la distribution dont l'échantillon est supposé être généré. Ce test est plus puissant que le précédent car c'est celui pour lequel le risque d'accepter H_0 à tort est plus faible.

Soit la fonction de répartition empirique

$$F_n(x) = \frac{1}{n} \sum_1^n 1_{X_i} \leq x. \quad (\text{I.6})$$

$$= \frac{\text{nombre d'observations } \leq x}{\text{la taille de l'échantillon } n}.$$

Et la statistique de décision est :

$$D = \sup_x | F_n(x) - F(x) |. \quad (\text{I.7})$$

On fixe un seuil de signification α et on compare D à des valeurs critiques particulières qu'on note D_n obtenues à partir de la table de Kolmogorov-Smirnov.

On accepte H_0 si $D < D_n$, sinon on rejette H_0 .

Exemple :

Soit un échantillon de taille $n = 4$ d'une population

0,34 – 0,49 – 0,56 – 0,70

x_i	0,34	0,49	0,56	0,70
$F_n(x_i)$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	1
$F(x_i)$	0,34	0,49	0,56	0,70
D_{x_i}	0,09	0,01	0,19	0,30

TABLEAU I.2- Résultats du test de kolmogorov-Smirnov.

On calcul $D = MAX D(x_i) = 0,30$.

La valeur lue sur la table de kolmogorov-Smirnov est $D_4 = 0,494$.

Comme $D < D_4$, on accepte H_0 , donc l'échantillon est bel et bien de la loi $\mathcal{U}_{[0,1]}$ [41].

I.2.8 Tests d'indépendance

Ce test est un test bilatéral utilisé pour vérifier si les observations sont mutuellement indépendantes.

On considère les données tel qu'elles sont collectées. On divise l'échantillon en deux classes :

n_1 : le nombre d'observation dans la classe une (1).

n_2 : le nombre d'observation dans la classe une (2).

On enregistre un signe (+) si l'observation est dans la classe (1).

On enregistre un signe (-) si l'observation est dans la classe (2).

On compte le nombre r de séquence homogènes de (+) et (-).

On définit un seuil de signification α et on accepte H_0 : "les observations sont indépendantes".

Si r est compris entre les valeurs critiques lues sur les tables 4 et 5 si n_1 et $n_2 \leq 20$ données en annexes .

Si n_1 ou $n_2 > 20$. On utilise le résultat suivant $r \rightsquigarrow \mathcal{N}(\mu, \sigma^2)$ ou $\mu = \frac{2n_1n_2}{n} + 1$ et $\sigma^2 = \frac{2n_1n_2(2n_1n_2 - n)}{n^2(n-1)}$.

Exemple :

Soit un échantillon de taille $n = 25$ (en utilisant les données du test de Khi-Deux).

On définit la classe 1 : de signe (+) pour les nombre ≤ 0.5 .

On définit la classe 2 : de signe (-) pour les nombre > 0.5 .

$n_1 = 16$, $n_2 = 9$, $n = n_1 + n_2 = 25$ et pour $r = 25$.

pour un seuil de signification $\alpha = 0.05$, les valeurs critiques lues sur les tables 4 et 5 sont

7 et 18, comme $r = 10$ est compris entre 7 et 18, on accepte l'hypothèse H_0 selon laquelle les observations sont mutuellement indépendantes[41].

I.2.9 Qualités d'un bon générateur

La réponse à la première question a donné lieu à une très abondante littérature. Les procédures qui permettent d'obtenir de telles suites de nombres sont totalement déterministes et plus ou moins sophistiquées. Voici la liste des qualités que devrait avoir un algorithme de génération de nombres pseudo-aléatoires définies par Brent.

- *Uniformité* : La suite doit passer avec succès les tests d'uniformité et d'indépendance précédents. Si de nombreux générateurs utilisés dans le passé avaient de très mauvaises propriétés statistiques, on dispose actuellement de générateurs qui passent convenablement ces tests.
- *Indépendance* : La suite $(u_n, n \geq 1)$ et aussi des sous-suites du type $(u_{nd}, n \geq 1)$ doivent être indépendantes au moins pour de "petites" valeurs de d , telles que $d \leq 6$.
- *Période* : La plupart des générateurs utilisés sont des suites périodiques et les programmes font facilement appel à de 10^n valeurs de la suite avec n de l'ordre de 30 ou plus. Ceci impose d'avoir un générateur ayant une très longue période.
- *Reproductibilité* : Pour tester un programme, il faut pouvoir reproduire exactement la suite de nombres $(u_n, n \geq 1)$ générée. A ce fin il est nécessaire d'associer un état (graine ou seed en anglais) au générateur afin de pouvoir rétablir une configuration déjà vue.
- *Portabilité* : Il faut pouvoir faire exécuter le programme sur des machines différentes et que les suites fournies par des ordinateurs avec une architecture de 32 bits ou de 64 bits soient identiques si elles ont la même valeur initiale.
- *Sous-suites disjointes* : Si une simulation est effectuée sur une machine multiprocesseurs ou si le calcul est distribué à travers un réseau, il faut que les sous-suites utilisées par chaque sous-tâche du programme soient indépendantes.
- *Efficacité* : L'appel au générateur étant fait un très grand nombre de fois, il faut que son programme soit le plus simple possible et nécessite peu d'opérations qui doivent être peu coûteuses en temps de calcul. La plupart des langages de programmation contient aujourd'hui un générateur de nombres pseudo-aléatoires[40].

I.2.10 Quelques générateurs de nombres pseudo-aléatoires

Il existe plusieurs algorithmes de génération des nombres pseudo-aléatoires. Citons quelques un :

Le générateur de Von Neumann

En 1946, John von Neumann propose un générateur pseudo-aléatoire connu sous le nom de la méthode middle-square (carré médian). Très simple, elle consiste à prendre un nombre, à l'élever au carré et à prendre les chiffres au milieu comme sortie. Celle-ci est utilisée comme graine pour l'itération suivante.

Exemple : prenons le nombre 1111. Les étapes de l'algorithme sont :

1. $1111^2 = 1234321$;
2. on récupère les chiffres du milieu : 3432. C'est la sortie du générateur ;
3. $3432^2 = 11778624$;
4. on récupère les chiffres du milieu : 7786, et ainsi de suite.

La méthode a surtout une importance historique. Elle est faible et la qualité des sorties dépend de la graine. Par exemple l'état 0000 produit toujours la même séquence et constitue un état absorbant de l'algorithme[15].

Le générateur Mersenne twister

Le Mersenne Twister est un générateur de nombres pseudo-aléatoires développé en 1997 par Makoto Matsumoto et Takuji Nishimura, est basé sur une matrice de récurrence linéaire sur un champ binaire finie. Il fournit pour une génération de nombres pseudo-aléatoires rapide et de très haute qualité, ayant été conçu spécifiquement pour corriger la plupart des défauts trouvés dans des algorithmes plus anciens. Le Mersenne Twister a été optimisé pour l'utilisation des simulations de Monte Carlo dans un certain nombre de domaines, y compris les voies biochimiques complexes et la finance. Le générateur Mersenne twister présente les propriétés suivantes :

- Sa période est de $2^{19937} - 1$;
- Il est uniformément distribué sur un grand nombre de dimensions ;
- Il est plus rapide que la plupart des autres générateurs (sauf les plus médiocres statistiquement) ;
- Il est aléatoire quel que soit le poids du bit considéré, et passe les tests Diehard.

Une révision de l'algorithme a été faite afin de combler quelques lacunes, notamment l'initialisation correcte, afin d'assurer la maximisation de la période[10].

Le générateur congruentiel

Le générateur congruentiel(en anglais congruential generator) a été proposée en 1951 par Lehmer pour générer des nombres pseudo-aléatoires selon une distribution uniforme

$\mathcal{U}_{[0,1]}$. C'est le type de générateurs le plus couramment mis en œuvre. L'état X_n à l'étape n est un entier et la fonction de transition est donnée par :

$$X_{n+1} = a.X_n + c \pmod{m} \quad n \geq 0. \quad (\text{I.8})$$

Où $a > 0, m > 0$ et c sont des entiers tels que :

a : multiplicateur.

m : modulo ou entier très grand.

c : constante additive.

X_0 : la valeur initiale.

Dans le cas où $c = 0$ le générateur est appelé Générateur Congruentielle Multiplicatif ou Linéaire (GCLM).

Dans le cas où $c \neq 0$ le générateur est appelé générateur congruentielle mixte.

La qualité d'un générateur à congruence linéaire dépend de façon cruciale du choix des constantes a, c et m .

Cette méthode est attractive d'un point de vue pratique, car lorsque un bon jeu de paramètres a été trouvé, le simulateur est extrêmement simple à mettre en œuvre et permet d'obtenir des nombres aléatoires de manière peu coûteuse en temps de calcul. De tels générateurs à congruence linéaire sont implémentés dans quasiment tous les langages de programmation et toutes les bibliothèques mathématiques [24, 41].

I.3 Simulation de variables aléatoires suivant une loi donnée

I.3.1 Méthode d'inversion

La méthode la plus simple s'applique lorsqu'on connaît une forme explicite de la fonction de répartition inverse F^{-1} . L'algorithme est le suivant :

- Générer un nombre aléatoire u uniforme dans $[0, 1]$;
- Retourner $X = F^{-1}(u)$.

1. La loi uniforme :

Pour simuler une variable aléatoire uniforme sur $[0, 1]$, il suffit de générer des nombres aléatoires \mathcal{U} d'une variable aléatoire qui suit une loi uniforme sur $[0, 1]$ et on déduit les réalisations suivantes :

$$X = (b - a)u + a. \quad (\text{I.9})$$

2. La loi exponentielle :

Soit X une variable aléatoire exponentielle, sa fonction de répartition est $F(x) = 1 - \exp - \lambda x$. La fonction inverse de F est :

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

Remarque : Si la variable aléatoire u est uniforme, $(1 - u)$ l'est aussi, on obtient une réalisation d'une variable aléatoire exponentielle, posant

$$X = -\lambda^{-1} \ln u [29]. \quad (\text{I.10})$$

I.3.2 Méthode de rejet

Si l'on désire générer des valeurs d'une variable aléatoire avec un densité $f(x)$, alors que ses propriétés ne sont pas connues, et sa fonction de répartition associée n'est pas inversible d'une manière explicite. La méthode de rejet est utilisée s'il existe une fonction $g(x)$ ayant un domaine identique à celui de f et une constante c positive.

1. Normaliser le domaine de $f(x)$ à l'aide d'une échelle c , de sorte que :

$$g(x) = c[f(x)] \leq 1, \quad a \leq X \leq b;$$

2. Définir X comme fonction linéaire de u :

$$X = a + (b - a)u;$$

3. Générer une paire de nombres aléatoires (u_1, u_2) ;
4. Chaque fois que l'on rencontre une paire de nombres aléatoires satisfaisantes :

$$u_2 \leq c[f(a + (b - a)u_1)]. \quad (\text{I.11})$$

On accepte $X = a + (b - a)u_1$ comme variable aléatoire suivant $f(x)$ [42].

I.3.3 Simulation de variable Gaussienne

La fonction de répartition d'une variable gaussienne (associée à la fonction d'erreur) n'est pas aisément calculable (elle peut être tabulée et approchée avec une grande précision toutefois). La méthode d'inversion de la fonction de répartition n'est donc pas aisée à mettre en oeuvre. Pour contourner cette difficulté, on peut utiliser une des deux méthodes suivantes[24] :

Méthode d'approximation

Cette méthode est basée sur le théorème central limite (I.2) :

Soient z_1, z_2, \dots, z_k k variables aléatoires indépendantes et identiquement distribuées telles que : $E(z_i) = m$ et $Var(z_i) = \sigma^2 \quad \forall i = \overline{1, k}$.

La variable aléatoire $y_i \rightsquigarrow \mathcal{N}(0, 1)$.

Cas particulier : Pour une loi des z_i uniforme sur $[0, 1]$ et $k = 12$, on a :

$$y_j = \sum_{i=1}^{12} z_i - 6$$

sont générées selon une loi normale centrée réduite.

Méthode de Box-Muller

La méthode de Box-Muller est plus rapide que la précédente et produit des couples de nombres aléatoires exactes selon une loi normale centrée réduite.

Soient X et Y deux v.a indépendantes qui suivent une loi normale centrée réduite.

Soit $f(x, y)$ la densité conjointe du couple (X, Y) :

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

On transforme x et y en coordonnées polaire :

$$x = r \sin \theta \quad y = r \cos \theta.$$

On remarque que :

- θ suit une loi uniforme sur $[0, 2\pi]$.
- $\frac{r^2}{2}$ suit une loi exponentielle de paramètre (1).

Algorithme de Box-Muller :

Algorithme I.1

- | |
|--|
| <ul style="list-style-type: none"> - Générer $u_1 \rightsquigarrow U_{[0,1]}$; - Poser $\theta = 2.\pi.u_1$; - Générer $u_2 \rightsquigarrow U_{[0,1]}$; - Poser $r = \sqrt{-2.\ln(1 - u_2)}$; - $x = r \sin \theta$; - $y = r \cos \theta$ [21]. |
|--|

I.4 Simulation de Monte Carlo

Le nom **Monte Carlo** (MC) a été proposé par les scientifiques du projet Manhattan lors de la deuxième guerre mondiale et fait allusion aux jeux de hasard pratiqués à

Monaco. Parmi les pionniers des méthodes MC nous retrouvons E. Fermi, J. Neumann, S. Ulam, N. Metropolis. Les méthodes MC sont aujourd'hui utilisées pour simuler des phénomènes physiques complexes dans plusieurs domaines scientifiques et appliqués : radioactivité, physique des hautes énergies, réseaux, économétrie, logistique. Il peut être visualisé comme une boîte noire où entre un flux de nombres pseudo-aléatoires (c-à-d générés par l'ordinateur) et on obtient un flux de nombres comme sortie ; l'estimation de la quantité d'intérêt est obtenue en analysant l'output[15].

La méthode Monte Carlo est une méthode d'approximation, au sens statistique du terme, elle représente les techniques permettant d'évaluer une quantité déterministe avec des paramètres ou des entrées stochastiques [12, 15].

I.4.1 Description de la méthode

L'idée sous-jacente aux méthodes de Monte-Carlo consiste à représenter la quantité que l'on cherche à calculer comme l'espérance d'une distribution et d'utiliser un générateur de nombre aléatoire pour construire un estimateur empirique de cette quantité[24], donc il faut calculer $\mathbb{E}(X)$. On dispose alors d'une suite $(x_i)_{1 \leq i \leq n}$ de n réalisations de la variable aléatoire X . On approxime alors $\mathbb{E}(X)$ par :

$$\mathbb{E}(X) \approx \frac{1}{n} \sum_{i=1}^n x_i \quad (\text{I.12})$$

Tel que X suit une loi uniforme sur $[0, 1]$ alors (I.12) devient comme suit :

$$\mathbb{E}(X) \approx \frac{1}{n} \sum_{i=1}^n u_i [12]. \quad (\text{I.13})$$

I.4.2 Simulation de Monte Carlo : calcul d'intégrale

L'intégrale d'une fonction $f(x)$ intégrable peut être exprimée comme la moyenne ou espérance de la fonction évalué à un emplacement aléatoire. Considérons une intégrale sur l'intervalle unité unidimensionnelle.

$$I = \int_0^1 f(x) dx = \bar{f} \quad (\text{I.14})$$

Soit x , une variable aléatoire qui est uniformément distribuée sur l'intervalle unité. Alors.

$$I = E[f(x)].$$

Pour une intégrale sur le cube unité $I^d = [0, 1]^d$ à d dimensions,

$$I = E[f(x)] = \int_{I^d} f(x) dx$$

dans laquelle x , est un vecteur uniformément répartie dans le cube unité.

La formule quadratique de Monte Carlo est basée sur l'interprétation probabiliste d'une intégrale. Considérons une suite $\{x_n\}$ échantillonnée à partir de la distribution uniforme. Une approximation empirique de l'espérance de f est :

$$I_n = \frac{1}{n} \sum_{i=1}^n f(x_i). \quad (\text{I.15})$$

Conformément à la Loi Forte des Grands Nombres (I.1), cette approximation est convergente presque sûrement :

$$\lim_{n \rightarrow +\infty} I_n = I$$

En outre, l'estimateur est sans biais, Pour tout n , ce qui signifie que la moyenne de I_n est exactement I ,

$$E[I_n] = I$$

Dans lequel la moyenne est indépendante du choix de l'échantillon $\{x_n\}$.

En général, on définit l'erreur d'intégration de Monte Carlo

$$\varepsilon_n = I - I_n$$

Alors le biais de l'estimateur est $E[\varepsilon_n]$, et la racine de l'erreur moyenne quadratique est : $\sqrt{E[\varepsilon_n^2]}$ [19].

La vitesse de convergence d'une telle méthode est fournie par le théorème Central Limite (I.2) qui est donc, en moyenne, en $o(N^{-\frac{1}{2}})$, indépendamment de la dimension d du problème. L'efficacité de l'estimateur I_n est alors $I/(t\sigma^2)$ où t est le temps de calcul pour obtenir I_n .

Exemple :

1. $I = \int_0^1 e^x dx$

Nous allons calculer cette intégrale en utilisant la suite de nombres aléatoires suivante : $U = (0.69 - 0.22 - 0.55)$ et on pose $f(x) = e^x$.

$$I_n = \frac{1}{n} \sum_{i=1}^n f(u_i)$$

$$= \frac{1}{3}(e^{0.69} + e^{0.22} + e^{0.55})$$

$$I_3 = 1.65.$$

Estimation la variance de $f(u)$:

$$\hat{\sigma}^2 = \widehat{Var}(f(u)) = \frac{1}{n} \sum_{i=1}^n f^2(u_i) - I_n^2.$$

$$\hat{\sigma}^2 = \frac{1}{3}(e^{2*0.69} + e^{2*0.22} + e^{2*0.55}) - (1.65)^2.$$

$$\hat{\sigma}^2 = 0.12.$$

$$\hat{\sigma} = 0.34.$$

Approximation de l'erreur $|I_n - I|$ à 95%.

$$\varepsilon = \hat{\sigma} \frac{1.96}{\sqrt{n}} = \frac{0.34 * 1.96}{\sqrt{3}}.$$

$$\varepsilon = 0.39.$$

$$2. I = \int \int_{\Delta} 1 \, dx \, dy.$$

Avec $\Delta = \{(x, y) \in \mathbb{R}^2 \text{ tel que } 0 \leq x \leq 1 \text{ et } 0 \leq y \leq \sqrt{1-x^2}\}$.

$$I = \int_0^1 dx \int_0^{\sqrt{1-x^2}} 1 \, dy = \int_0^1 dx (y \Big|_0^{\sqrt{1-x^2}}).$$

$$= \int_0^1 \sqrt{1-x^2} \, dx.$$

On pose

$$x = r \cos \theta$$

$$r = [0, 1].$$

$$y = r \sin \theta$$

$$\theta \in [0, \frac{\pi}{2}].$$

$$\int \int_{\Delta} dx \, dy = \int \int |J| \, dr \, d\theta = \int_0^1 \int_0^{\frac{\pi}{2}} dr \, \theta \Big|_0^{\frac{\pi}{2}} = \frac{1}{2} \frac{\pi}{2} (r \Big|_0^1) = \frac{\pi}{4}.$$

On a $f^*(x, y) = \{1 \text{ si}$.

$$u_1 = (0.58 - 0.26) \in \Delta.$$

$$u_2 = (0.78 - 0.26) \notin \Delta.$$

$$u_3 = (0.7 - 0.58) \in \Delta.$$

$$u_4 = (0.26 - 0.74) \in \Delta.$$

$$u_5 = (0.56 - 0.54) \in \Delta.$$

$$u_6 = (0.04 - 0.16) \in \Delta.$$

$$I_n = I_6 = \frac{1}{6} \sum_{i=1}^n f^*(u_i).$$

$$I_6 = \frac{1}{6}(5) = 0.83.$$

$$Q = \text{Aire}(R) * I_6 = 1 * 0.83 = 0.83.$$

Approximation de l'erreur $|Q - I|$ à 95% est :

$$Q = \text{Aire}(R) * \hat{\sigma} \frac{1.96}{\sqrt{n}} = \frac{\hat{\sigma} * 1.96}{\sqrt{6}}.$$

$$\hat{\sigma} = \sqrt{\frac{1}{n} \sum_{i=1}^n f^{*2}(u_i) - I_n^2}.$$

$$\sqrt{0.83 - (0.83)^2} = 0.37.$$

$$\text{Par conséquent } E = \frac{0.37 * 1.96}{\sqrt{6}} = 0.29 \text{ [42].}$$

I.4.3 Erreurs dus à la méthode de MC

Lors de l'utilisation de l'échantillonnage aléatoire simple (SRS) ou l'approche Monte Carlo, deux types d'effets sont présents dans un échantillon aléatoire - celui lié à l'ensemble des valeurs qui représente l'effet ensembliste et celui à leur séquences (séquentiel).

- **Effet ensembliste** : (erreurs indésirables)

Ce sont des erreurs dues à l'ensemble des valeurs produites par la loi uniforme de la méthode de MC. Ceci est caractérisé par la différence entre l'échantillon généré et la distribution théorique.

- **Effet séquentiel** : (erreurs désirables)

Ce sont des erreurs dues à la suite de cet ensemble de valeurs. Ceci est caractérisé par l'ordre dans lequel les valeurs de la loi uniforme se présentent.

Remarque : Il existe deux méthodes qui réduisent les erreurs d'échantillonnage.

I.4.4 La méthode replicated runs (histoires répliquées)

Notre souci est de produire de bons estimateurs des variables de sortie, pour cela plusieurs histoires indépendantes de simulation sont nécessaires, et cela est une façon de réduire la variance de la variable de sortie [42].

I.4.5 Réduction de la variance

Une méthode pour rendre plus efficace un estimateur est de réduire sa variance (sans augmenter significativement le temps de calcul). Plusieurs techniques réduisant la variance, mais pas la vitesse de convergence (excepté pour la variante du cas de l'échantillonnage stratifié), sont présentées maintenant. Il est aussi possible de combiner plusieurs techniques de réduction de la variance, mais une telle pratique doit être réalisée avec prudence car elle ne produit pas toujours les effets escomptés : en effet, la variable aléatoire dont on cherche l'espérance n'est alors plus la même. Afin d'illustrer les méthodes, nous nous limiterons au calcul d'une intégrale $I = \int_{[0,1]^d} f(x)dx$ [54].

Variables de contrôle

L'erreur d'approximation dans l'estimation de I par une méthode de Monte Carlo standard provient des grandes variations de f . La technique de variable de contrôle réécrit

l'intégrale sous la forme :

$$I = \int_{[0,1]^d} g(x)dx + \int_{[0,1]^d} (f(x) - g(x))dx \quad (\text{I.16})$$

Si g est suffisamment simple pour être intégrée formellement, on peut se concentrer sur la deuxième intégrale. De même, si g est suffisamment proche de f (si g absorbe les variations de f), on obtient une réduction conséquente de la variance. En effet, si X désigne une variable aléatoire uniformément distribuée sur $[0, 1]^d$ et si l'on pose $Z = f(X) - g(X)$, on a :

$$\sigma^2(Z) = \sigma^2(f(X)) + \sigma^2(g(X)) - 2Cov(f(X), g(X))$$

Pour obtenir une réduction de la variance, il suffit donc que :

$$\sigma^2(g(X)) - 2Cov(f(X), g(X)) < 0 \quad (\text{I.17})$$

Variation antithétiques

Les variables antithétiques aussi appelées variables complémentaire reposent sur ce principe :

Faire une exécution en utilisant les nombres aléatoires (u_1, \dots, u_n) pour obtenir un échantillon d'observations aléatoires (x_1, \dots, x_n) puis calculer l'estimateur $\hat{\theta}$.

faire une autre exécution en utilisant les nombres aléatoires complémentaires $(1-u_1, \dots, 1-u_n)$ pour obtenir un échantillon d'observations aléatoires (x'_1, \dots, x'_n) puis calculer l'estimateur $\hat{\theta}'$.

Utiliser ces deux exécutions pour calculer la moyenne combinée : $\hat{\theta}_{nac} = \frac{\hat{\theta} + \hat{\theta}'}{2}$ qui est un estimateur sans biais et plus précis que $\hat{\theta}$ [42].

Stratification

L'idée ici est de séparer le domaine d'intégration en plusieurs sous-domaines, et de réaliser un échantillonnage de type Monte Carlo standard sur chacun de ces sous-domaines[22].

La première étape consiste à l'estimation de $\theta = \mathbb{E}[f(X)]$,

puis simuler n_j points $(X_{1j}, \dots, X_{n_j j})$ dans chaque Ω_j avec probabilité $P(\cdot | \Omega_j)$. L'estimateur est :

$$\hat{\theta} = \sum_{j=1}^J \frac{P(\Omega_j)}{n_j} \sum_{i=1}^{n_j} f(X_{ij}).$$

La variance estimée est :

$$\hat{\sigma}(\hat{\theta})^2 = \sum_{j=1}^J \frac{P(\Omega_j)^2 \hat{\sigma}_j^2}{n_j}, \hat{\sigma}_j^2 = \frac{1}{n_j} \sum_{i=1}^{n_j} f(X_{ij}^2) - \left(\frac{1}{n_j} \sum_{i=1}^{n_j} f(X_{ij}) \right)^2.$$

Échantillonnage préférentiel

C'est une méthode de calcul de $\theta = \mathbb{E}[f(X)]$ qui consiste à tirer les variables selon une distribution erronée, et à compenser numériquement le résultat à postériori.

Pour en donner une idée, revenons au calcul d'une intégrale sur $[0, 1]$. Si p est une densité strictement positive sur $[0, 1]$, on peut écrire

$$\int_0^1 f(x)dx = \int \frac{f(x)}{p(x)}p(x)dx \quad (\text{I.18})$$

Ce qui suggère la possibilité d'un autre algorithme : $\hat{\theta}' = \frac{1}{n} \sum_{i=1}^n \frac{f(Y_i)}{p(Y_i)}$.

Où les Y_i sont i.i.d sur $[0, 1]$ de loi $p(x)dx$.

Rappelons que si X a une densité $f_X(x)$ et Y une densité f_Y , alors $p = f_Y/f_X$ [22].

I.5 Les méthodes non aléatoires

La méthode de Monte Carlo avec échantillonnage aléatoire est la méthode de base pour calculer l'intégrale. Bon nombre de méthodes alternatives à la méthode de Monte Carlo ont été proposées pour améliorer la convergence, parmi lesquelles les méthodes de simulation pseudo-probabilistes, comme l'échantillonnage stratifié ou par Hypercube Latin (LHS), les méthodes de Quasi-Monte Carlo, ou encore les méthodes de Quasi-Monte Carlo Randomisé[31].

I.5.1 L'échantillonnage Descriptif(ED)

Proposé comme une approche alternative à la simulation de Monte Carlo, l'échantillonnage descriptif, (Descriptif Sampling DS)(Saliby, 1980 et 1990) est basé sur une sélection totalement déterministe des valeurs d'échantillonnage d'entrée et de leurs permutations aléatoire.

En tant que tel, l'échantillonnage descriptif évite l'effet ensembliste du à la méthode de MC et conduit à des estimations plus précises de simulation[48].

Principe :

- Calculer $r_i = \frac{i-0.5}{n}$ $i = \overline{1, n}$ où r_i est le milieu de chaque sous intervalle de $[0, 1]$.
- Calculer les valeurs de l'échantillon x_i qui sont obtenues par la méthode d'inversion.
- Mélanger aléatoirement les observations de l'échantillons $x_i, i = \overline{1, n}$.
- Stocker la suite obtenue et l'utiliser lorsque c'est nécessaire.

Les avantages :

Retrait de l'effet ensembliste.

Représente mieux que la méthode de MC la variable d'entrée.

Les inconvénients :

Cette méthode a été critiquée pour son biais de l'échantillonnage qu'elle peut produire. la taille de l'échantillon doit être connue au début de la simulation.

Exemple :

Estimer la moyenne de la loi exponentielle de paramètre ($\lambda = 1$), $r_i = \frac{i-0.5}{n}$.

Soit $n = 4$.

i	r_i	x_i
1	0.125	2.079
2	0.375	0.981
3	0.675	0.471
4	0.875	0.133
Total		3.664

TABLE I.1 – Résultats de l'exemple de la méthode ED

L'estimateur de la moyenne est :

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{3.664}{4} = 1.22.$$

I.5.2 L'échantillonnage descriptif amélioré (RDS)

Pour réduire le risque du biais de la méthode précédente, une approche a été proposée, qui consiste en un bloc qui doit être situé à l'intérieur d'un générateur visant à distribuer aléatoirement des sous-ensembles réguliers de tailles de nombre premiers P_q, P_{q+1}, \dots pour tout q dans un ordre aléatoire à la demande de la simulation. Nous nous arrêtons de produire ces sous suites lorsque la simulation s'arrête (le critère d'arrêt est la durée de la simulation dans une exécution).

Dans cette approche, chaque exécution est déterminée par un bloc de différents nombres premiers. Si on a besoin de M exécutions répétées, il est nécessaire de considérer M blocs de sous-ensembles réguliers m_1, m_2, \dots, m_M .

Les nombres premiers et les valeurs de la variable aléatoire des sous-ensembles ne sont pas les mêmes pour toutes les exécutions répliquées.

Principe :

- Générer un nombre premier P_1 puis calculer les sous suites de nombres $r_i = \frac{i-0.5}{P_1}$ $i = \overline{1, P_1}$.
- Mélanger les r_i .
- Générer un deuxième nombre premier P_2 et calculer $r_i = \frac{i-0.5}{P_2}$.

Les avantages :

Elle garde les avantages de la méthode précédente de plus cette approche élimine la nécessité de déterminer à l'avance la taille de l'échantillon.

Les estimateurs sont sans biais.

Exemple :

Simuler la loi exponentielle avec $\mathbb{E}(X) = 1$ tel que $r_i = \frac{i-0.5}{P} \rightsquigarrow \mathcal{U}_{[0,1]}$, $i = \overline{1, P}$.

Avec la méthode d'inversion on obtient :

$$x_i = -\ln(1 - r_i) \quad i = \overline{1, P} \quad \text{et on estime sa moyenne.}$$

On génère $P_1 = 7$, $P_2 = 11$ et $P_3 = 13$.

i	r_i	x_i	r_i	x_i	r_i	x_i
1	0.071	0.071	0.045	0.047	0.038	0.039
2	0.214	0.241	0.136	0.147	0.115	0.123
3	0.357	0.442	0.227	0.248	0.192	0.214
4	0.5	0.693	0.318	0.383	0.269	0.314
5	0.643	1.030	0.409	0.526	0.346	0.425
6	0.786	1.540	0.5	0.693	0.423	0.550
7	0.927	2.639	0.591	0.894	0.5	0.693
8		$T = 6.659$	0.682	1.145	0.577	0.860
9			0.773	1.482	0.654	1.061
10			0.864	1.992	0.731	1.312
11			0.955	3.091	0.808	1.649
12				$T=10.647$	0.885	2.159
13					0.962	3.258
						$T = 12.662$

TABLE I.2 – Résultats de l'exemple de la méthode RDS

Nous calculons les trois moyennes \overline{y}_1 , \overline{y}_2 et \overline{y}_3 .

$$\overline{y}_1 = \frac{6.659}{7} = 0.951.$$

$$\overline{y}_2 = \frac{10.648}{11} = 0.968.$$

$$\overline{y}_3 = \frac{12.662}{13} = 0.974.$$

L'estimation de la moyenne en utilisant cette méthode est obtenue par :

$$\hat{y} = \frac{1}{3} \sum_{i=1}^3 \bar{y}_i = 0.965[50].$$

I.5.3 L'échantillonnage par Hypercube Latin

L'échantillonnage par hypercube latin est une technique de criblage, dans lequel la sélection de valeurs d'échantillon est fortement contrôlée, bien que toujours en leur permettant de varier[48].

L'échantillonnage par hypercube latin (Latin HyperCube Sampling) (LHS) est un cas particulier d'échantillonnage stratifié où on s'assure que le découpage a défini des sous-espaces équiprobables, et que chaque sous-espace est échantillonné de manière uniforme. Plus précisément, chaque intervalle de chacune des variables est partitionné en n sous-intervalles de même probabilité $[\frac{i-1}{n}, \frac{i}{n}]$, $i = \overline{1, n}$. Une valeur est sélectionnée de manière uniforme dans chaque sous-intervalle. Les n valeurs obtenues pour la variable X^1 sont couplées aux n valeurs de la variable X^2 de manière aléatoire et équiprobable pour chaque combinaison.

Plus précisément, on se donne deux permutations aléatoires uniformes indépendantes π_1 et π_2 de $\{1, \dots, n\}$. La valeur de X^1 sur le sous-intervalle $[\frac{\pi_1(i-1)}{n}, \frac{\pi_1(i)}{n}]$ sera couplée à la valeur de X^2 sur le sous-intervalle $[\frac{\pi_2(i-1)}{n}, \frac{\pi_2(i)}{n}]$. Ces n paires sont ensuite combinées de manière aléatoire et équiprobable avec les n valeurs de la variable X^3 pour former n triplets, et ainsi de suite jusqu'à ce que n d-uplets soient formés. On peut donc générer un LHS de n points en dimension d , en prenant d permutations aléatoires uniformes indépendantes de $\{1, \dots, n\}$, $\pi_j(1), \dots, \pi_j(n)$, $j = \overline{1, d}$. Les d coordonnées des n points sont alors données par :

$$x_i^j = \frac{\pi_j(i) - U_i^j}{n} \quad j = \overline{1, d}, i = \overline{1, n}. \quad (\text{I.19})$$

Où U_i^j sont nd variables aléatoires i.i.d de loi $\mathcal{U}_{[0,1]^d}$. La figure (I.3) illustre un échantillonnage par hypercube latin en dimension $d = 2$. Sur cette figure, on voit que l'intervalle de chaque variable a été découpé en $n = 10$ sous-intervalles de même taille. Pour chaque variable et dans chaque sous-intervalle un point a été généré selon une loi uniforme.

Les couples (X^1, X^2) ont été obtenus à l'aide des permutations $\pi_1 = (5; 4; 7; 9; 8; 2; 10; 3; 1; 6)$ et $\pi_2 = (9; 4; 5; 7; 10; 6; 8; 1; 2; 3)$.

L'estimateur de l'intégrale I est donné par la même formule que l'estimateur Monte Carlo

$$I_n^{LHS} = \frac{1}{n} \sum_{i=1}^n h(x_i). \quad (\text{I.20})$$

où pour $i = \overline{1, n}$, x_i est issu d'un échantillonnage par hypercube latin.

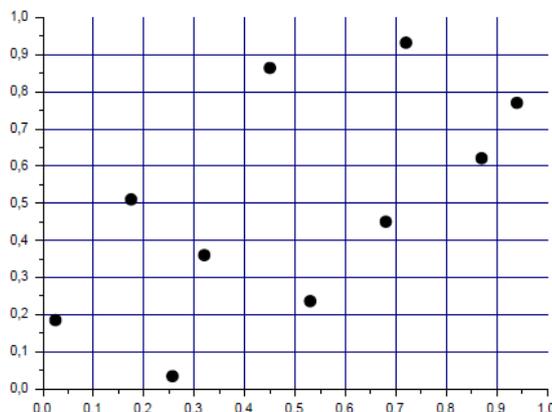


FIGURE I.3 – Représentation graphique d'un ensemble de points d'un échantillonnage par LHS

On peut prouver que lorsque la fonction h est monotone en chacune de ses variables, alors la variance de l'estimateur I_n^{LHS} où les points sont échantillonnés à l'aide d'un hypercube latin, est inférieure à la variance de I_n^{MC} . Cependant, l'hypothèse de monotonie est, en pratique, rarement satisfaite : on dispose alors d'un résultat asymptotique, qui garantit, lorsque h est de carré intégrable et lorsque $n \rightarrow \infty$, une diminution de la variance de l'estimateur I_n^{LHS} par rapport à celle de I_n^{MC} . De plus, cette diminution est d'autant plus importante que h est proche d'une fonction additive[55].

I.5.4 L'échantillonnage par Supercube Latin

L'échantillonnage par Supercube latin (Latin Supercube Sampling)(LSS) est utilisée pour des simulations de dimension très élevés, tels que ceux survenant dans le transport de particules et les files d'attente. LSS est développé comme une combinaison de deux méthodes couramment utilisés : (LHS) et quasi-Monte Carlo (QMC). En LSS, les variables d'entrée sont regroupées en sous-ensembles, et la méthode QMC dont la dimension est moins élevée est utilisé a l'intérieur de chaque sous-ensemble. Les points de QMC sont présentés dans un ordre aléatoire à l'intérieur des sous-ensembles[44].

I.5.5 Quasi Monte Carlo

Les méthodes de Quasi-Monte Carlo sont des versions déterministes des méthodes de Monte Carlo. Ces méthodes définissent des séquences d'échantillons déterministes qui

ont une discrédance plus faibles que les séquences aléatoires, c'est-à-dire qu'elles ont une meilleure répartition uniforme dans l'espace des variables d'entrée. Ces méthodes permettent d'obtenir une convergence plus rapide en $o(N^{-1}(\log N)^d)$ (sous des conditions relativement faibles de régularité de f). Parmi les séquences utilisées, celles de Halton, séquences de Sobol ou encore de Faure peuvent être citées[15].

I.5.6 Quasi Monte Carlo randomisées

Les méthodes de Quasi-Monte Carlo Randomisées, sous certaines conditions peu restrictives sur f , ont un taux de convergence en $o(N^{-\frac{3}{2}}(\log N)^{-\frac{p-1}{2}})$ et permettent une approximation de l'erreur d'estimation[40].

Le principe consiste à prendre les séquences déterministes a_i de ces dernières, et on les transforme en variables aléatoire x_i . Cette transformation se fait par exemple par[31] :

$$x_i = a_i + X. \tag{I.21}$$

I.6 Conclusion

Les méthodes de Monte Carlo ont comme grand avantage, par rapport aux méthodes d'analyse numérique classiques, d'avoir une vitesse de convergence en $o(\frac{1}{\sqrt{n}})$ (pour un échantillon de n points), donc indépendante de la dimension du problème. Néanmoins, il doit exister des suites de nombres telles que la convergence soit plus rapide, en supprimant l'aspect aléatoire. Ceci conduit à s'intéresser aux méthodes dites de Quasi Monte Carlo[53].

CHAPITRE II

Les méthodes de Quasi Monte Carlo

II.1 Introduction

Dans ce chapitre on se place dans un espace A de dimension d . On cherche à estimer l'intégrale d'une fonction inconnue f définie sur A . La première idée serait d'utiliser la méthode de Monte Carlo standard. Cependant, il est fréquent que pour des raisons de coût, la taille de l'échantillon des réalisations de f soit limitée. La précision obtenue n'est alors plus toujours satisfaisante. La méthode de Quasi Monte Carlo (QMC) ayant une meilleure vitesse de convergence que Monte Carlo standard, pour un même nombre de points, la précision devrait être meilleure. Cependant, cette vitesse de convergence dépend de la dimension. Plus précisément, lorsque la dimension est trop élevée, la précision de Monte Carlo est en théorie meilleure que celle de Quasi Monte Carlo[5].

Nous discutons dans ce chapitre des méthodes alternatives à celles de Monte Carlo connues sous le nom de méthode de Quasi Monte-Carlo. Ces méthodes diffèrent des méthodes de Monte Carlo classique dans le sens où elles ne sont pas basées sur la génération de suites pseudo-aléatoires mais sur l'intégration sur des ensembles de points engendrés de façon déterministe et possédant des propriétés fortes d'équirépartition[24].

II.2 Définition de la méthode Quasi Monte Carlo

Une des façons d'améliorer les méthodes de type Monte Carlo est de renoncer au caractère aléatoire des tirages et de trier les points de façon "plus ordonnée", c-à-d d'essayer de retirer les erreurs liées à l'ensemble des valeurs générés par la méthode de MC . On cherche à trouver des suites $P_n = (x_i)_{i \geq 0}$ déterministes permettant d'approximer des intégrales, en utilisant la loi forte des grands nombres on obtient :

$$\int_{[0,1]^d} f(x)dx \approx \lim_{n \rightarrow +\infty} \frac{1}{n} (f(x_1) + \dots + f(x_n)) [36] \quad (\text{II.1})$$

Où $P_n \subset [0, 1]^d$, est choisi de façon plus uniformément distribué qu'un ensemble de points aléatoire. On parle dans ce cas de méthode de Quasi Monte Carlo[2].

Les méthodes QMC peuvent être décrites comme des versions déterministes des méthodes de Monte Carlo, dans le sens où les échantillons aléatoires dans les méthodes de Monte Carlo sont remplacés par des points déterministes bien choisis. L'intérêt particulier des méthodes QMC porte sur les mesures d'uniformité de la distribution et sur le développement des suites bien réparties[56].

II.3 Quelques notions utilisées sur les méthodes QMC

II.3.1 La discrédance

Soit un ensemble de points $x_1, \dots, x_n \subset I^d$, et un sous-ensemble $G \subset I^d$. On définit la fonction $S_n(G)$ par le nombre de points $x_i \in G$, $\forall i = \overline{1, n}$. Pour chaque $\gamma = (\gamma_1, \dots, \gamma_d) \in I^d$, nous introduisons une région rectangulaire $G_\gamma = [0, \gamma_1) \times \dots \times [0, \gamma_d)$ avec le volume d-dimensionnel $\gamma_1 \dots \gamma_d$. La discrédance des points x_1, \dots, x_n est donnée par :

$$D(x_1, \dots, x_n) = \sup_{\gamma \in I^d} \left| \frac{S_n(G_\gamma)}{n} - \gamma_1 \dots \gamma_d \right| [38]. \quad (\text{II.2})$$

Une autre définition simple et basique de la discrédance est donnée pour comprendre en quoi elle répond à nos attentes de remplissage de l'espace.

La discrédance mesure l'écart entre une distribution de points uniforme et une distribution de points donnée; en d'autres termes elle permet de mesurer l'irrégularité de la distribution.

Si $d = 1$ on considère \hat{F}_n la fonction de répartition empirique des points x_1, \dots, x_n , on définit la discrédance par :

$$D_n(X) = \sup_{x \in [0,1]} \left| \hat{F}_n(x) - F_U(x) \right| \quad (\text{II.3})$$

F_U est la fonction de répartition de la loi uniforme sur $[0, 1]$ [25].

Il existe de nombreux résultats de majoration de la discrédance. Le résultat le plus fameux est connu sous le nom d'inégalité de Koksma-Hlawka[30] aussi appelé théorème de bornes qui donne une définition précise de l'erreur absolue lié à la discrédance. Mais avant définissons la variation de Hardy et krause.

II.3.2 La variation de Hardy Krause

La variation totale de f au sens de Hardy et Krause est :

$$V(f) = \sum_{k=1}^d \sum_{i_1 < i_2, \dots, < i_k} \int_{[0,1]^d} \left| \frac{\partial^k f}{\partial x_{i_1} \dots \partial x_{i_k}} \right| dx_{i_1} \dots dx_{i_k} \quad (\text{II.4})$$

où f est calculée en $x_j = 1$ pour les j différents de i_1, \dots, i_k . On a le théorème de Koksma et Hlawka[22].

II.3.3 Inégalité de Koksma-Hlawka

Théorème II.1

Pour toute suite x_n et une fonction de variation bornée, l'erreur d'intégration est défini par :

$$\left| \frac{1}{n} \sum_{i=1}^n f(x_i) - \int_{[0,1]^d} f(t) dt \right| \leq V(f) D_n(X) \quad (\text{II.5})$$

Ce qui est égale à :

$$\varepsilon_n = I[f] - I_n[f] \leq V(f) D_n(X) \quad (\text{II.6})$$

Où $V(f)$ est la variation Hardy-Krause à la dimension d de la fonction f .

Remarque : Lorsque $V(f)$ est fini, la formule (II.6) montre qu'une erreur liée à l'estimation de f peut être calculée à partir d'une mesure de discrédance sur l'ensemble de points.

II.3.4 Suites à faible discrédance

Discrédance et divergence sont deux mesures bien connues de l'irrégularité de la distribution des ensembles de points, elles jouent un rôle majeur dans les méthodes QMC pour choisir de bons ensembles de points. Le critère de discrédance est particulièrement adapté pour l'intégration numérique, et le critère de divergence est particulièrement utile dans l'étude des problèmes d'optimisation globale[56].

Les suites à faible discrédance sont des suites déterministes ayant de meilleures propriétés d'uniformité que les suites aléatoires dans le cube de dimension d , dont la discrédance est d'ordre $o(\log n)^d n^{-1}$ pour n assez grand. La définition de ces suites est tirée de l'inégalité de Koksma-Hlawka.

Une suite à faible discrédance doit fournir une estimation $I_n[f]$ de $I[f]$, de sorte que l'erreur ε_n soit bornée par $o((\log n)^d n^{-1})$ [46]. Ces suites comprennent les suites de Van der corput[23], Halton[27], Sobol[49], Hammersley[28], et les suites de Faure[38].

II.3.5 La fonction radical inverse

Definition II.1

Soit un entier $n \geq 2$, la fonction radicale inverse en base b est donnée par :

$$\phi_b(n) = \frac{a_0}{b} + \frac{a_1}{b^2} + \dots + \frac{a_m}{b^{m+1}}. \quad (\text{II.7})$$

Où a_i est la position du bit en partant de droite, $\forall i = \overline{0, m}$.

avec : $n = a_0 + a_1b + \dots + a_mb^m$

$$\text{et : } m = \begin{cases} 1 + \lfloor \log_b(n) \rfloor & \text{si } n \neq 0; \\ 1 & \text{sinon.} \end{cases} \quad [25]$$

II.3.6 Les nombres de directions

Definition II.2

Un ensemble de nombres de direction v_i est défini comme étant

$$v_i = \frac{m_i}{2^i}, i = \overline{1, w} \quad (\text{II.8})$$

Les m_i peuvent être choisis librement à condition que chaque m_i soit impair et inférieur à 2^i [32].

Par exemple, $n = 5$ est 101 dans la base 2, on peut prendre $v_1 = 0.1$, $v_2 = 0.11$, $v_3 = 0.111$ [14].

II.3.7 Le code Gray

Definition II.3

Le code de Gray a été proposé par Antonov et Saleev[11], il est défini comme suit :

$$G(j) = j \oplus \left\lfloor \frac{j}{2} \right\rfloor = (\dots j_3 j_2 j_1)_2 \oplus (\dots j_4 j_3 j_2)_2 \quad (\text{II.9})$$

où : \oplus désigne l'addition modulo, appelé aussi le "ou" exclusif ou XOR[32].

II.3.8 Le polynôme primitif

Definition II.4

Soit x une variable aléatoire et b_i , $i = \overline{1, l-1}$ les coefficients prenant pour valeurs 0 ou 1.

Un polynôme primitif est défini comme suit :

$$P(x) = x^l + b_1 x^{l-1} + b_2 x^{l-2} + \dots + b_{l-1} x + 1 \quad (\text{II.10})$$

Avec le $\text{pgcd}(b_i) = 1$ et $P(x)$ est irréductible sur \mathbb{Z}_2 [14].

II.3.9 La matrice de Pascal

Une matrice génératrice de Pascal est $C(c_{kl})$ est donnée par :

Definition II.5

$$C_{kl} = \binom{l-1}{k-1} = \begin{cases} \frac{(l-1)!}{(k-1)!(l-k)!} & \text{si } k \leq l, \\ 0 & \text{sinon.} \end{cases}$$

$\forall l, k \in \mathcal{N}^*$

II.4 Etat de l'art

Les méthodes de Quasi Monte Carlo, encore appelées méthodes d'équidistribution ont été étudiées depuis leur apparition dans les travaux de Koksma (1942/1943) et Hlawka (1964). Elles ont pour origine des résultats issus de la théorie des nombres appliqués à l'analyse numérique[33]. Le nom quasi-Monte Carlo fut employé pour la première fois dans un rapport de recherche en 1951 de Richtmyer. Roth, médaillé Fields en 1958, a déterminé en 1954 une vitesse de convergence optimale pour l'approximation des intégrales, ainsi qu'une suite, utilisant l'idée de Van der Corput, permettant une convergence rapide. Ce sont alors développées au cours des années les suites de QMC dites à faible discrèpance basées sur les théorèmes de bornes[1].

La première suite concrète à faible discrèpance a été explicitement donnée par Halton en 1960[37]. Ces suites forment une généralisation multidimensionnelle des suites en dimension 1 de Van der Corput[54].

Braaten et Weller (1979)[18] ont amélioré les suites de Halton en prenant des permutations qui minimisent la discrèpance de chaque suite de Van Der Corput en base b , Cette méthode ne permet pas d'avoir une seule permutation qui implique une meilleure discrèpance. Ainsi une table de permutations est fournie jusqu'à la dimension 16. Kocis et Whiten (1997)[34] proposent aussi deux méthodes pour améliorer les suites de Halton. Tuffin (1998)[52] s'est également intéressé au choix des permutations en particulier pour les dimensions plus élevées[25].

En 1960, la suite de Hammersley en dimension d est construite à partir d'un terme dépendant du nombre de points et d'une suite de Halton en dimension $(d-1)$ [25].

Plus tard, en 1967 les suites de Sobol ont été proposées à l'aide de permutations de la suite de Van der Corput en base 2[24], l'inconvénient de ces suites réside dans le fait

que l'on ne peut pas ajouter de nouveaux points, pour contourner ce problème, Sobol et Levitan en 1976 proposent une table de valeurs jusqu'en dimension 16 de manière à avoir des distributions de points plus uniformes. Une solution pour conjurer ce problème est d'utiliser des méthodes de scrambling[25].

En 1982, les suites de Faure ont été définies à partir de la matrice de Pascal.

Korobov (1959) et Hlawka (1962) sont les premiers qui ont introduit l'ensemble des réseaux et qui ont prouvé l'existence d'un tel ensemble avec une faible discrédance, que l'on appelle Good lattice points[25], en 1987 Sloan et Kachoyan ont généralisé la construction ci-dessus[57].

Owen[43] présente les méthodes de Quasi Monte Carlo randomisées (Randomized Quasi Monte Carlo(RQMC)) comme une randomisation des séquences utilisées dans les méthodes de quasi-Monte Carlo.

II.5 Les suites de QMC

II.5.1 Les suites de Van Der Corput

Nous introduisons dans cette section une classe spécifique de suite à discrédance faible en dimension 1, appelée suite de **Van der Corput**.

Soit n un entier, tout entier positif n possède une représentation unique (appelée représentation en base b) comme une combinaison linéaire de puissances positives de b à coefficient à valeurs dans $\{0, \dots, b-1\}$

$$n = \sum_{j=0}^{\infty} a_j(n, b) b^j. \quad (\text{II.11})$$

Cette décomposition comporte un nombre fini de termes non nuls. La fonction radicale inverse définie en (II.1) $\phi_b(n)$ transforme n en un point dans $[0, 1]$ qui est appelée suite de Van Der Corput en base b donnée par :

$$\phi_b(n) = \sum_{j=0}^{\infty} a_j(n, b) b^{-j-1} [24]. \quad (\text{II.12})$$

La discrédance de la suite est donnée par :

$$D_n^*(\phi) = O\left(\frac{\ln(n)}{n}\right) \quad [39]. \quad (\text{II.13})$$

La taille d'un cycle varie selon les différentes bases, la quantité des nombres qui remplissent l'intervalle $[0, 1]$ dans chaque cycle représente la taille du cycle. En base 2, les paires

$(0, \frac{1}{2}), (\frac{1}{4}, \frac{3}{4})$ sont les deux premiers cycles.

Par exemple, en base 3, la suite a le chiffre 3 comme puissance dans le dénominateur avec un cycle de longueur trois $C = 3$ (par exemple, $0, \frac{1}{3}, \frac{2}{3}$ est le premier cycle).

La suite est donnée par :

$$0, \frac{1}{3}, \frac{2}{3}, \frac{1}{9}, \frac{4}{9}, \frac{7}{9}, \frac{2}{9}, \frac{5}{9}, \frac{8}{9}, \frac{1}{27}, \frac{10}{27}, \frac{19}{27}, \dots [3].$$

Exemple

Supposons $b = 3$ et $n = 19$, on peut écrire 19 en base 3 comme suit :

$19 = 2 * 3^2 + 0 * 3^1 + 1 * 3^0 = 201$, donc la suite de Van Der Corput s'écrit comme :

$$\phi_3(19) = \frac{1}{3} + \frac{0}{9} + \frac{2}{27} = \frac{11}{27} [35].$$

II.5.2 Les suites de Halton

Les suites de Halton sont les plus élémentaires des suites à faible discrédance multi-dimensionnelles, ce qui peut être considéré comme la pierre angulaire d'autres suites à faible discrédance. La suite de Halton est une suite générale d -dimensionnelle dans le cube unité $[0, 1]^d$. La première dimension de la suite de Halton est la suite de Van der Corput en base 2 et la seconde dimension de Halton est la suite de van der Corput utilisant une base 3. La dimension d de la suite de Halton est le $d^{\text{ème}}$ nombre premier, qu'utilise la suite de Van Der Corput comme base. Plus la base de la suite de Van Der Corput augmente, plus la dimension croit. La suite prend plus de temps pour remplir le cube unité.

Par exemple : pour $d = 25$ et $d = 26$ les nombres premiers correspondants respectivement sont $b = 97$ et $b = 101$ [35].

Soit $b_1, \dots, b_d \in \{2, 3, 4, \dots\}$, la suite de Halton en dimension d est :

$$\phi_{b_1, \dots, b_d} = (\phi_n : n \in \{0, 1, 2, \dots\}). \quad (\text{II.14})$$

Où $\phi_n = (\phi_{b_1}(n), \dots, \phi_{b_d}(n))$ [26].

La discrédance de la suite de Halton est donnée par :

$$D_n^*(\phi_b^d) \leq \frac{1}{n} \prod_{i=1}^d \frac{b_i \ln(b_i n)}{\ln(b_i)} = O\left(\frac{(\ln(n))^d}{n}\right) [39]. \quad (\text{II.15})$$

En dimension $d = 1$ d'une base b , le $n^{\text{ème}}$ élément de la suite $\{\phi_n\}$ est obtenu en suivant ces étapes :

1. Ecrire le nombre n dans la base b ,

2. Inversez les chiffres et mettre un point radical (c'est-à-dire une base décimale b) en face de la suite,
3. Le résultat est $\phi_b(n)$.

Par exemple, supposons $b = 3$ et $n = 22$:

1. $22 = 2 * 3^2 + 1 * 3^1 + 1 * 3^0 = 211$,
2. Nous obtenons 0.112 base 3.

Exemples

1. Les 10 premiers nombres de la suite de Halton dans les trois premières dimensions sont donnés dans le tableau suivant suivi de leurs représentations graphique pour $d = 1(b = 2)$ et $d = 2(b = 3)$.

	<i>Dim = 1 (Base 2)</i>	<i>Dim = 2 (Base 3)</i>	<i>Dim = 3 (Base 5)</i>
$n = 1$	1/2	1/3	1/5
$n = 2$	1/4	2/3	2/5
$n = 3$	3/4	1/9	3/5
$n = 4$	1/8	4/9	4/5
$n = 5$	5/8	7/9	1/25
$n = 6$	3/8	2/9	6/25
$n = 7$	7/8	5/9	11/25
$n = 8$	1/16	8/9	16/25
$n = 9$	9/16	1/27	21/25
$n = 10$	5/16	10/27	2/25

TABLE II.1 – Les 10 premiers nombres de la suite de Halton

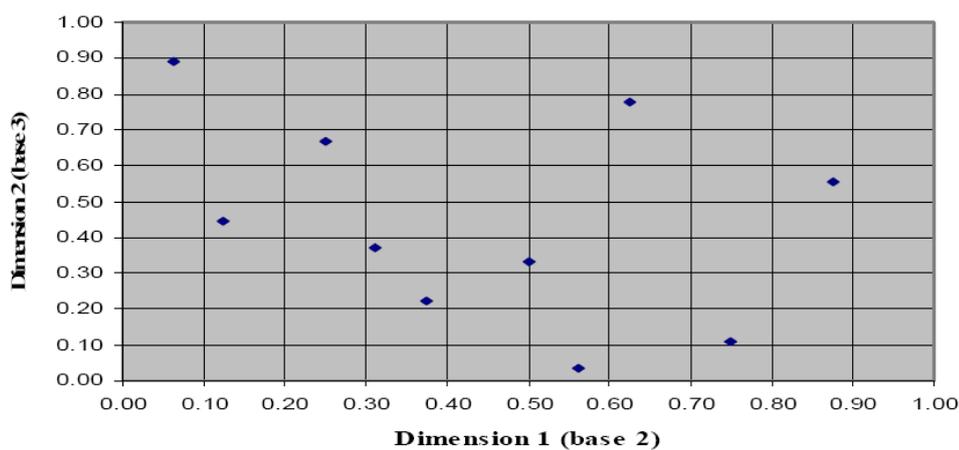


FIGURE II.1 – Représentation graphique des 10 premiers nombres de la suite de Halton

2. Les deux figures suivantes représentent la projection des points de la suite de Halton dans le cube unité pour $n = 1000$ tel que la figure (II.2) est représentée avec $d = 14(b = 43)$ et $d = 15(b = 47)$.

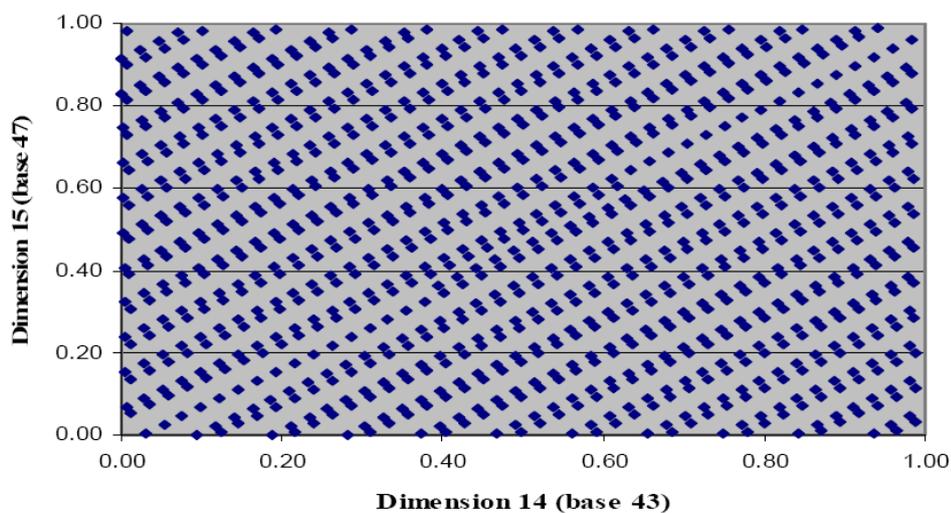


FIGURE II.2 – Les 1000 nombres de la suite de Halton en dimension $14 * 15$

la figure (II.3) montre toujours les suites de Halton tel que $d = 27(b = 103)$ et $d = 28(b = 107)$ [35].

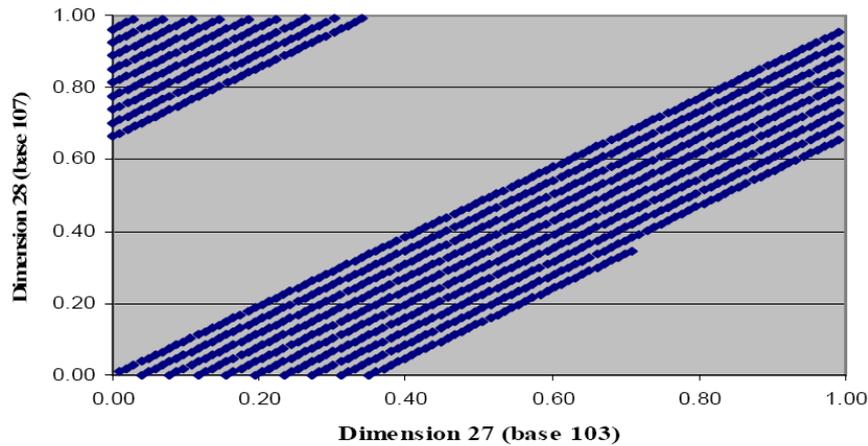


FIGURE II.3 – Les 1000 nombres de la suite de Halton en dimension $27 * 28$.

Malgré l'uniformité de la distribution dans le cube unité (II.2), ces suites montrent une certaine faiblesse qui se pose dans les dimensions élevées.

En effet, lorsque l'on projette les points d'une telle suite dans le carré unité (II.3), nous pouvons observer des diagonales décalées à chaque itération, qui laissent place à de nombreuses zones lacunaires dans le carré unité. Ce phénomène résulte de la régularité des sous-suites de longueur b des suites de Van der Corput[25].

Le problème avec les suites de Halton réside du fait que les points dans les dimensions successives sont fortement corrélées, et en dimension élevée, les points initiaux sont regroupés près du point $(0,0)$. Tous ces problèmes conduisent à de faibles estimations d'intégrales[35].

Amélioration

Une solution pour corriger cette pathologie est de casser les corrélations entre les suites de Van Der Corput des différentes dimensions. Braaten et Weller ont amélioré l'uniformité des points d'une suite de Halton en appliquant sur chaque nombre premier b_j une permutation π_{b_j} sur $\{0, \dots, b_j - 1\}$ tel que $\pi_{b_j}(0) = 0$. Ainsi, à partir du nouveau développement de n en base b_j :

$$\phi_{b_j}(n) = \frac{\pi_{b_j}(a_0)}{b_j} + \frac{\pi_{b_j}(a_1)}{b_j^2} + \dots + \frac{\pi_{b_j}(a_m)}{b_j^{m+1}}. \quad (\text{II.16})$$

On obtient une suite de Halton $\phi_{b_1, \dots, b_d} = \{x_0, x_1, \dots, x_{n-1}\}$ tel que $x_i = (\phi_{b_1}(i), \dots, \phi_{b_d}(i))$ dite généralisée. La démarche est aussi dénommée comme une méthode de Scrambling.

Ces suites permettent d'améliorer l'uniformité de la répartition mais elles ont des caractéristiques encore pas très connues[25].

II.5.3 Les suites de Hammersley

Une suite Hammersley en dimension d est construite à partir d'un terme dépendant du nombre de points et d'une suite de Halton en dimension $(d - 1)$ [25].

Soit $b_1, \dots, b_{d-1} \in \{2, 3, 4, \dots\}$ des entiers positifs premiers entre eux deux à deux. L'ensemble de points N de Hammersley en dimension $d > 2$ est :

$$\phi(n) = \left(\frac{n}{N}, \phi_{b_1}(n), \dots, \phi_{b_{d-1}}(n) \right); \quad n = 0, \dots, N - 1. \quad (\text{II.17})$$

La discrédance de la suite de Hammersley est donnée par :

$$D_N^*(\phi_b^d) \leq c_d \frac{(\log N)^{d-1}}{N} [37]. \quad (\text{II.18})$$

Ces suites étant construites à partir de suites de Halton, elles présentent le même phénomène de diagonales successives. De plus, il est impossible de rajouter des points supplémentaires à ces suites sans pour autant perturber la discrédance. Dans le cas où le nombre de points à générer n'est pas connu à l'avance, il est déconseillé d'utiliser une suite de Hammersley. Aussi, ces suites ont perdu l'aspect itératif des suites de Halton qui permettait de rajouter des points facilement par contre l'avantage des ensembles de Hammersley sur les suites de Halton est le taux de convergence de la borne pour la discrédance est légèrement plus rapide car elle dépend de $(d - 1)$ base au lieu de d) [25, 26].

Exemple Voici la représentation graphique d'ensembles de points ($n = 500$) de la suite de Hammersley dans les bases 2, 3, 7 et 11

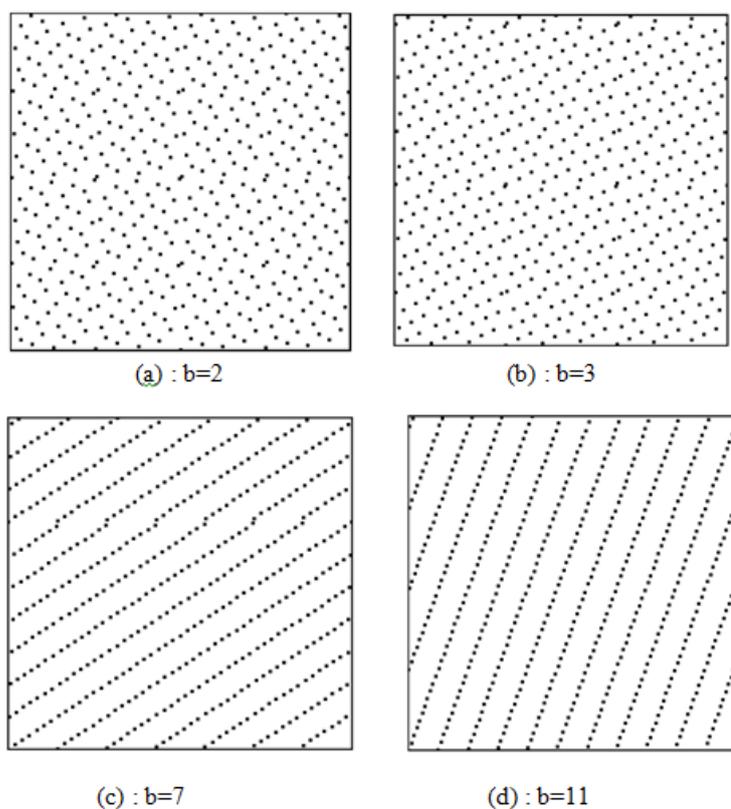


FIGURE II.4 – Les 500 nombres de la suite de Hammersley.

La figure (II.4) montre la distribution de la suite de Hammersley avec des bases différentes lorsque $n = 500$.

On remarque que quand $b = 2$ (II.4(a)), on obtient le modèle le plus uniformément réparti, dès que la base augmente (II.4(b),(c),(d)) le modèle devient de plus en plus régulier. Les points ont tendance à s'aligner en formant des diagonales [58].

II.5.4 Les suites de Sobol

La suite de Sobol est l'une des suites à faible discrédance standard et la plus utilisée dans les applications quasi-Monte Carlo, il a été prouvé qu'elle a une propriété d'uniformité supplémentaire sous certaines conditions d'initialisation[20]. Les suites de Sobol de dimension d utilisent le nombre premier 2 comme base. La suite de Sobol en $d = 1$ est la suite de van der Corput en base 2, et dans les grandes dimensions, la suite est présentée comme des permutations de la suite en première dimension[35].

Sa construction est basée sur les polynômes primitifs dans le domaine \mathbb{Z}_2 définis en (II.4) et les opérations XOR. Les permutations générées à partir de ces polynômes irréductibles permettent de développer certaines corrélations, alors elles peuvent produire des régions

où aucun point ne peut s'y trouver jusqu'à ce que n devient très grand[7].

Les permutations de la suite de Sobol dépendent des nombres de direction définis en (II.2).

Une suite de Sobol en dimension 1 est définie comme suit :

$$\phi(n) = a_1v_1 \oplus a_2v_2 \oplus \dots \oplus a_wv_w[14]. \quad (\text{II.19})$$

On reprend l'exemple présenté dans la définition (II.2) pour $n = 5$, on a :

$$\phi(5) = 1 * 0.1 \oplus 0 * 0.11 \oplus 1 * 0.111.$$

$$= 0.1 \oplus 0 \oplus 0.111$$

$$\phi(5) = 0.011$$

La construction la suite de Sobol pour une dimension d nous procédons comme suit :

1. Choisir un entier n au hasard, par exemple, $n = 2$. Ce nombre définit le point de départ de la suite.
2. Calculer le Code Gray (II.3) de n , $G(n)$.
3. Transformer $G(n)$ à sa représentation binaire ($G(2) = 3 = 1 * 2^1 + 1 * 2^0 = 11$).
4. Sommer bit par bit les nombres de direction associé aux chiffres de $G(n)$ dans la base binaire, dans cet exemple on compte de droite à gauche les bits de $G(n)$ différent de zéro qui sont v_1 et v_2 , par conséquent $y(2; d) = 0.1 \oplus 0.11 = 0.01$.
5. Transformer le nombre obtenu en un nombre décimal en multipliant chaque bit de cette représentation binaire par 2^l , où l est la position du bit dans la partie décimale du nombre, et sommer de gauche à droite les termes. Le résultat est le premier terme de la suite de Sobol en dimension d :

$$\phi_2^d(n) = 0 * 2^{-1} + 1 * 2^{-2} = 0.250[35].$$

Amélioration

Une amélioration des suites de Sobol a été proposée par Antonov et Saleev[11] qui peut être générée beaucoup plus rapidement tout en conservant de bonnes propriétés de convergence. Dans leur définition, la suite des nombres de directions v_i est générée par le polynôme primitif avec les coefficients appartenant au corps \mathbb{Z}_2 , On définit la suite $v(d)$ de la manière suivante :

$$m(k) = 2b_1m_{k-1} \oplus 2^2b_2m_{k-2} \oplus \dots \oplus 2^{l-1}b_{l-1}m_{k-l+1} \oplus 2^l m_{k-l} \oplus m_{k-l} \quad (\text{II.20})$$

Où l est le degré du polynôme choisi, b_i les coefficients du polynôme[14].

Supposons que nous avons un polynôme primitif $x^3 + x + 1$. ($b_1 = 0$ et $b_2 = 1$), l'équation récursive des nombres de direction donne : $m(k) = 4m(k-2) \oplus 8m(k-1) \oplus m(k)$

à ce stade, nous supposons que les valeurs d'initialisation $m_1 = 1$, $m_2 = 3$ et $m_3 = 7$, nous calculons $m_4, m_5 \dots$ etc en utilisant l'équation récursive ci-dessus.

$m_4 = 12 \oplus 8 \oplus 1 = 5$, $m_5 = 28 \oplus 24 = 37$, et ainsi de suite, après avoir calculé les nombres d'initialisation nous allons utilisé la formule ref pour calculer les nombres de direction, on a :

$v_1 = 0,5$, $v_2 = 0,75$, $v_3 = 0,875$, $v_4 = 0,3125$ et $v_5 = 0,21875$.

Le $j^{\text{ème}}$ élément de la suite Sobol peut être obtenu à partir du $(j-1)^{\text{ème}}$ avec un seul nombre de direction : $y(j, d)_2 = y(j-1, d)_2 \oplus v_{jd}$.

La discrédance de la suite de Sobol est donnée par :

$$D_n^*(\phi_2^d) \leq c_d \frac{(\log n)^d}{n} + o\left(\frac{(\log n)^{d+1}}{n}\right) \quad (\text{II.21})$$

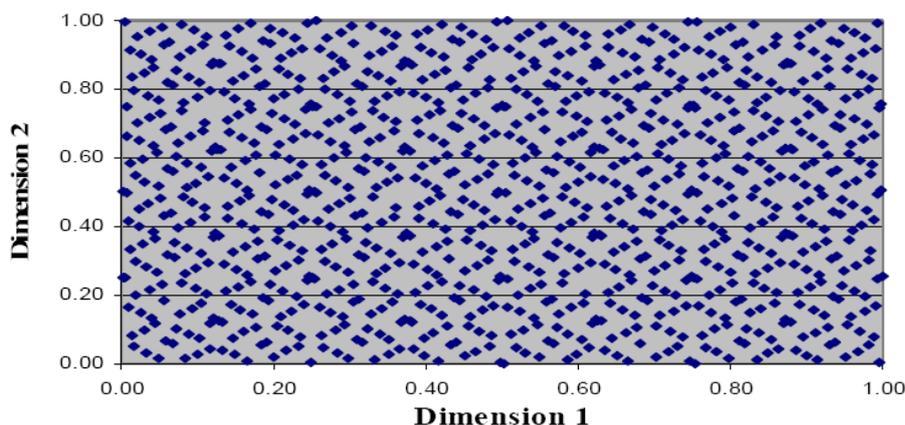
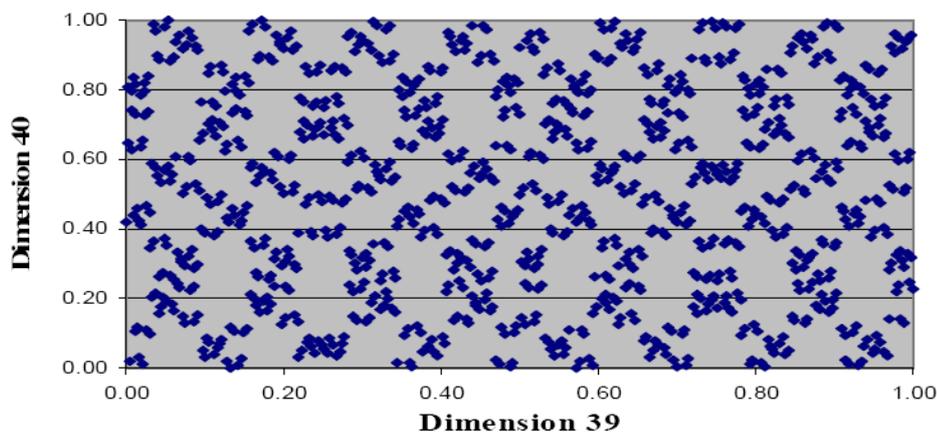
Exemple

1. Ce tableau illustre la suite de Sobol dans les 3 premières dimensions.

	<i>Dim</i> = 1 (Base 2)	<i>Dim</i> = 2 (Base 2)	<i>Dim</i> = 3 (Base 2)
$n = 1$	1/2	1/2	1/2
$n = 2$	1/4	1/4	3/4
$n = 3$	3/4	3/4	1/4
$n = 4$	3/8	3/8	5/8
$n = 5$	7/8	7/8	1/8
$n = 6$	5/8	1/8	3/8
$n = 7$	1/8	5/8	7/8
$n = 8$	3/16	5/16	5/16

TABLE II.2 – La suite de Sobol dans les 3 premières dimensions.

2. Représentations graphique des 1000 nombres de la suite de Sobol tel que la figure (II.5) est en $d = 1$ et $d = 2$ et la figure (II.6) est en $d = 39$ et $d = 40$ [35].

FIGURE II.5 – Les 1000 nombres de la suite de Sobol en dimension $1 * 2$ FIGURE II.6 – Les 1000 nombres de la suite de Sobol en dimension $39 * 40$.

Ces suites ont l'avantage d'être rapides à construire car la nature binaire des ordinateurs permet de réduire le temps de calcul. Elles ont aussi l'avantage de préserver généralement l'uniformité de la distribution lorsque la dimension augmente. Cependant, en raison de la nature arbitraire des états initiaux l_1, \dots, l_d de ces récurrences, les points sont souvent mal répartis dans l'hypercube unité. Seulement, on ne peut difficilement prédire les états initiaux qui donnent des distributions non uniformes. Un autre inconvénient apparaît si l'on souhaite rajouter de nouveaux points à une distribution. Dans ce cas, on observe qu'ils tombent dans le voisinage de ceux produits précédemment. Augmenter le nombre de points n'est donc pas une solution envisageable[25].

II.5.5 Les suites de Faure

L'uniformité des suites de Halton et Hammersley se dégradent en grande dimension parce que les éléments sont définies à l'aide de suite de Van der Corput associées à des bases $b \geq 1$. Faure a défini différentes extensions des suites multi-dimensionnelles de Van der Corput dans lesquelles les mêmes bases sont utilisées pour tout les éléments. Cette base b doit être un nombre premier supérieur ou égal à la dimension de l'espace, mais ce nombre est typiquement beaucoup plus petit que le $d^{\text{ème}}$ nombre premier utilisé pour définir les suites de Halton et Hammersley. Dans la méthode de Faure, les éléments sont construits en permutant des sous-suites des suites de Van der Corput et sont définies à partir de la fonction radicale inverse, ϕ_b , et une matrice génératrice de Pascal qui est définie en (II.5).

Comme base b , nous choisissons le plus petit nombre premier plus grand que d . Comme précédemment, nous construisons le $n^{\text{ème}}$ élément de la suite, le premier élément est donné par $\phi_b(n)$. Le $i^{\text{ème}}$ élément, $i = \overline{1, d}$, du $n^{\text{ème}}$ point de la suite de Faure est donné par :

$$\phi_b^i(n) = \sum_{j=1}^{\infty} y_j^{(i)}(n) b^{-j} \quad (\text{II.22})$$

où :

$$y_j^i = \sum_{l=j-1}^{\infty} \binom{l}{j-1} (i-1)^{l-j+1} a_l(n) \pmod{b}$$

Les sommes (II.11) et (II.22) comportent un nombre fini de termes non nuls. Si le développement de n en base b possède exactement r termes non nuls.

$$\begin{cases} a_l(n) = 0 & \text{si } l \geq r, \\ a_{r-1}(n) \neq 0 & \text{sinon} \end{cases}$$

alors $y_j(n) = 0 \forall j \geq r$. De façon matricielle :

$$\begin{pmatrix} y_1^{(i)}(n) \\ y_2^{(i)}(n) \\ \dots \\ y_r^{(i)}(n) \end{pmatrix} = C^{i-1} \begin{pmatrix} a_0(n) \\ a_1(n) \\ \dots \\ a_{r-1}(n) \end{pmatrix} \quad (\text{II.23})$$

où la matrice C^i est une matrice $r \times r$ dont les éléments sont donnés par :

$$\begin{cases} C^i(m, n) = \binom{n-1}{m-1} i^{n-m} & \text{si } n \geq m, \\ C^i(m, n) = 0 & \text{sinon} \end{cases}$$

Remarque :

Les matrices $C^{(i)}$ vérifient la propriété suivante :

$$C^{(i)} = C^{(1)}C^{(i-1)}, \quad i = 1, 2, \dots$$

Cette relation implique en particulier que :

$$\begin{pmatrix} y_1^{(i+1)}(n) \\ y_2^{(i+1)}(n) \\ \dots \\ y_r^{(i+1)}(n) \end{pmatrix} = C^{(1)} \begin{pmatrix} y_1^{(i)}(n) \\ y_2^{(i)}(n) \\ \dots \\ y_r^{(i)}(n) \end{pmatrix}.$$

Ce qui permet d'exprimer (II.23) de façon équivalente :

$$y_j^{(i+1)}(n) = \sum_{l=j-1}^{\infty} \binom{l}{j-1} y_l^{(i)}(n) \pmod{b} \quad (\text{II.24})$$

Pour comprendre les effets de la soustraction entre (II.11) et (II.22), considérons les entiers $0, \dots, b^r - 1$.

La représentation b -aire de ces entiers possède au plus r termes. Pour $n \in \{0, \dots, b^r - 1\}$, les vecteurs $a(n) = [a_0(n), a_1(n), \dots, a_{r-1}(n)]^T$ parcourent l'ensemble des b_r vecteurs de dimension r de coordonnées $\{0, \dots, b^r - 1\}$.

Les vecteurs $C^{(i)}a(n) \pmod{b}$, $n \in [0, b_r[$ parcourent le même ensemble qui peut être vu comme une permutation de $a(n)$.

La même propriété reste vraie pour $n \in \{jb^r, \dots, (j+1)b^{r-1}\}$, $j \in [0, b-1]$.

La discrédance de la suite de faure est donnée par :

$$D_n^*(\phi^d) \leq \frac{1}{n} \left[\left(\frac{b-1}{2 \log b} \right)^d \log^d(N) + o(\log(N)) \right]. \quad (\text{II.25})$$

Exemple

Pour illustrer la construction précédente, considérons le cas $r = 2$ et $b = 3$. Les matrices $C^{(1)}$ et $C^{(2)}$ sont données par :

$$C^{(1)} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \quad C^{(2)} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$$

Pour $k = 1, \dots, 8$, les vecteurs $a(n)$ sont donnés par :

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

Les vecteurs $C^{(1)}a(n) \bmod b$ sont donnés par :

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Les vecteurs $C^{(2)}a(n) \bmod 2$ sont donnés par :

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

En multipliant ces vecteurs par $(1/3, 1/9)$ et en réarrangeant ces fractions en trois lignes, nous obtenons les 9 premiers éléments de la suite de Faure en dimension 3[24].

n	Dim=1 (Base 3)	Dim=2 (Base 3)	Dim=3 (Base 3)
$n = 1$	$1/3$	$1/3$	$1/3$
$n = 2$	$2/3$	$2/3$	$2/3$
$n = 3$	$1/9$	$4/9$	$7/9$
$n = 4$	$4/9$	$7/9$	$1/9$
$n = 5$	$7/9$	$1/9$	$4/9$
$n = 6$	$2/9$	$8/9$	$5/9$
$n = 7$	$5/9$	$2/9$	$8/9$
$n = 8$	$8/9$	$5/9$	$2/9$

TABLE II.3 – La suite de Faure dans les 3 premières dimensions.

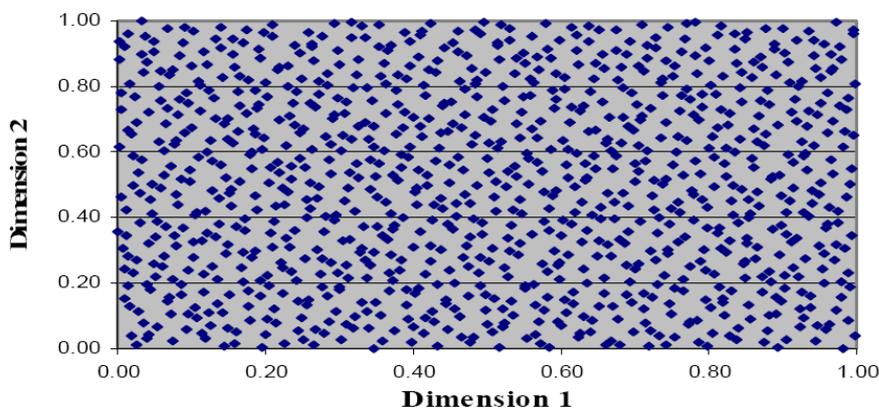


FIGURE II.7 – Les 1000 nombres de la suite de Faure en dimension $1 * 2$

Les problèmes des suites de Faure surviennent avec les dimensions élevées car elles exigent un nombre premier assez grand comme base et provoquent plus de corrélation entre les composantes voisines de la suite[35].

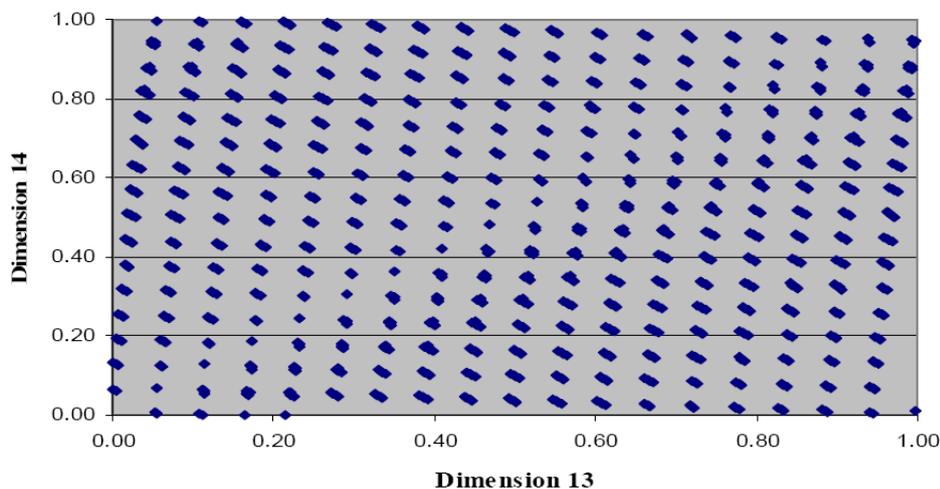


FIGURE II.8 – Les 1000 nombres de la suite de Faure en dimension $13 * 14$.

II.5.6 Good lattice points

Les good lattice points sont la forme la plus simple des méthodes de réseaux. Les good lattice points peuvent être considérés comme des suites à faible discrédance qui présentent des propriétés analogues à celles des grilles régulières telles que l'espacement régulier entre les points.

P est un "good lattice points" s'il est défini par :

$$P = \left(\left\{ \frac{i - 0,5}{n} \right\}, \left\{ \frac{g_2 i - 0,5}{n} \right\}, \dots, \left\{ \frac{g_d i - 0,5}{n} \right\} \right), \quad j = 1, \dots, d \text{ et } i = 1, \dots, n \quad (\text{II.26})$$

où $\{x\} = x - [x]$ est la partie fractionnaire de x , $(n; g_1, \dots, g_d)$ un vecteur d'entiers satisfaisant $1 \leq g_j \leq n, g_k \neq g_j$ pour $k \neq j$, et $d < n$ avec $g_1 = 1$.

Le terme "good" se réfère à un choix judicieux de n et des g_j . En effet, les générateurs sont déterminés de façon à minimiser la discrédance comme le montre cette figure.

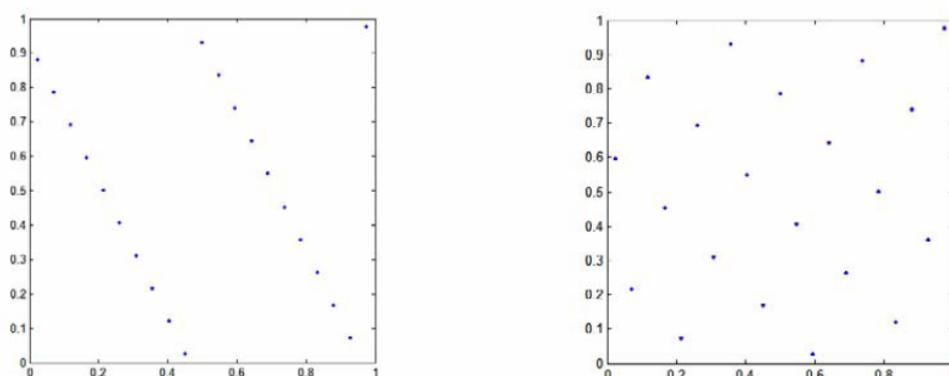


FIGURE II.9 – Les 21 premiers termes d'un good lattice points en dimension 2.

La figure (II.9) représente un good lattice points en dimension 2 avec $n = 21$ points. Sur la figure de gauche, nous avons pris des générateurs quelconques $g_1 = 1$ et $g_2 = 19$. Tandis que sur la figure de droite, les générateurs ont été choisis à l'aide de la table trouvée dans Fang et al. (1994) et valent $g_1 = 1$ et $g_2 = 13$.

Un good lattice point est en réalité un hypercube latin qui présente une faible discrédance. Ceci a pour avantage d'exclure les hypercubes latins dont les points sont mal répartis comme par exemple ceux dont les points se situent sur la diagonale. Cependant, cela n'évite pas les alignements comme l'illustre la Figure où les points sont positionnés sur 6 droites parallèles. L'avantage des good lattice points est donc leur régularité qui constitue également un inconvénient à cause des alignements obtenus. Une autre difficulté est celle d'ajouter des points à la distribution préexistante[25].

II.5.7 Quasi Monte Carlo randomisées

L'évaluation numérique d'une intégrale basée sur les suites à faible discrédance n'est pas une méthode aléatoire, par conséquent, il n'y a pas d'intervalle de confiance autour de l'estimation comme cela est le cas pour l'estimateur de Monte Carlo, au mieux, il y a une borne pour l'erreur mais l'erreur réellement commise peut être beaucoup plus petite. La marge d'erreur associée à l'intervalle de confiance (Monte Carlo) et la borne pour l'erreur (Quasi Monte Carlo) ne sont pas des quantités comparables, afin de pouvoir calculer une marge d'erreur dans le cas Quasi Monte Carlo, il est possible de randomiser les suites à faible discrédance[26].

L'utilisation de techniques de randomisation permet d'employer le théorème central limite pour déterminer/estimer l'erreur dans les méthodes QMC. Elles peuvent être aussi vues comme des techniques de réduction de la variance dans les méthodes Monte Carlo, en tant que généralisation des variables antithétiques. Une technique, appelée décalage aléatoire,

consiste à ajouter une même variable aléatoire X uniformément distribuée sur $[0, 1]^d$ à chaque élément de la suite afin de conserver la bonne répartition de cette dernière. Si on s'intéresse à :

$$Z = \frac{1}{N} \sum_{i=1}^N f(\{X + \phi(n)\}) \quad (\text{II.27})$$

au lieu de $\frac{1}{N} \sum_{n=1}^N f(\phi(n))$ dans QMC, où $\{x\}$ désigne le vecteur des parties fractionnaires des coordonnées de x , l'idée est de calculer la valeur moyenne obtenue sur I copies indépendantes Z_i de Z donnée par :

$$\hat{I} = \frac{1}{I} \sum_{i=1}^I Z_i \quad (\text{II.28})$$

et d'obtenir un intervalle de confiance en utilisant le théorème central limite sur ces I variables. La bonne répartition de la suite à faible discrédance doit permettre de réduire la variance de l'estimateur c-à-d qu'il devienne sans biais.

Pour appeler la fonction f le même nombre de fois pour chaque méthode, nous devons comparer la variance de l'estimateur (II.28) avec celle de l'estimateur Monte Carlo standard (I.15) pour N variables aléatoires.

Nous obtiendrons une réduction de la variance si et seulement si :

$$\sigma^2 \left(\frac{1}{N} \sum_{n=1}^N f(\{X + \phi(n)\}) \right) \leq \frac{1}{N} \sigma^2(f(X)) \quad (\text{II.29})$$

Il est possible d'obtenir une réduction conséquente de la variance par comparaison avec celle de la somme de N variables aléatoires indépendantes et identiquement distribuées $f(X)$, sous certaines hypothèses (par exemple si f est à variation bornée) :

$$\sigma^2 \left(\frac{1}{N} \sum_{n=1}^N f(\{X + \phi(n)\}) \right) = o(n^{-2} (\ln n)^{2d}) \quad (\text{II.30})$$

La convergence est donc plus rapide. En pratique, cette vitesse de convergence est aussi valable pour d'autres fonctions que celles à variation bornée[54].

II.6 Conclusion

Dans ce chapitre, nous avons étudié la méthode d'intégration numérique de Quasi-Monte Carlo (QMC) qui peut être considérée comme une version déterministe de la méthode de Monte Carlo. Nous avons donné les fondements et propriétés théoriques de l'approche Quasi Monte Carlo. Elle repose sur un échantillonnage déterministe du do-

maine d'intégration de la fonction étudiée par des suites de points uniformes, appelées suites à faible discrédance. L'utilisation de ces suites garantit un taux de convergence asymptotique théorique d'un ordre minimal, ce qui est nettement meilleur que le taux de convergence de la méthode de Monte Carlo. Le principal inconvénient de la méthode de Quasi-Monte Carlo est qu'il n'est pas possible d'estimer l'erreur d'intégration commise à partir des simulations réalisées, comme c'est le cas dans la méthode de Monte Carlo.

La méthode QMC perd de son efficacité lorsque la dimension du problème augmente, contrairement à la méthode de Monte Carlo dont la vitesse de convergence ne dépend que du nombre de simulations réalisées[45].

CHAPITRE III

Application et comparaison

III.1 Introduction

Théoriquement, la génération des nombres aléatoires suivant une loi donnée se ramène à la génération des suites de variables aléatoires indépendantes de loi uniforme sur $[0, 1]$. Dans ce chapitre nous allons procéder à l'application numérique de quelques suites à faible discrédance, pour cela nous allons utiliser les résultats obtenus comme un flux de nombres pseudo-aléatoires entrants dans la simulation de variables aléatoires suivant une loi donnée.

Dans notre cas, l'application se portera sur la génération d'ensembles de points engendrés de façon déterministe avec les suites de Halton, Sobol et Faure comme entrée dans la simulation de variables aléatoires suivant une loi exponentielle de paramètre λ .

La programmation de ces suites et la simulation de la variable aléatoire exponentielle se fera avec le logiciel MATLAB.

En premier lieu, nous procéderons aux tests d'ajustement à la loi uniforme puis à la simulation d'une variable aléatoire suivant une loi exponentielle à l'aide de la méthode d'inversion, enfin nous procéderons à la comparaison entre les trois méthodes qui sont QMC, MC et RDS par le moyen de deux critères de comparaison l'erreur absolue et l'erreur moyenne en anglais (Mean Square Error)(MSE).

III.2 Présentation du logiciel MATLAB

MATLAB (matrix laboratory) est un langage de programmation de quatrième génération et un environnement de développement ; il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet la manipulation de matrice, afficher des courbes et des données, mettre en œuvre des algorithmes, créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran. Les utilisateurs de MATLAB (environ un million en 2004) sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que la recherche. Matlab peut s'utiliser seul ou bien avec des toolbox (boîte à outils)[6].

III.3 Génération des suites de QMC

III.3.1 La suite de Halton

Nous avons généré des échantillons de taille $n = 10$ de la suite de Halton en base 2, 7 et 17, puis $n = 1000$ en les bases 17, 41 et 179, en utilisant la formule (II.14).

les résultats obtenus sont résumés dans le tableau suivant :

n	1	2	3	4	5	6	7	8	9	10
$\phi_2(n)$	0.5	0.25	0.75	0.125	0.625	0.375	0.875	0.0625	0.5625	0.3125
$\phi_7(n)$	0.1429	0.2857	0.4286	0.5714	0.7143	0.8571	0.0204	0.1633	0.3061	0.4490
$\phi_{17}(n)$	0.0588	0.1176	0.1765	0.2353	0.2941	0.3529	0.4118	0.4706	0.5294	0.5882

TABLE III.1 – La suite de Halton en base 2, 7 et 17.

Nous avons procédé à une série de tests afin de vérifier si nos échantillons suivent bien la loi uniforme sur $[0, 1]$, pour cela nous avons utilisé le test d'ajustement de Kolmogorov-Smirnov, tel que la valeur critique lue sur la table de Kolmogorov-Smirnov pour $n = 10$ est $D_n = 0.410$ pour $\alpha = 0.05$.

Les statistiques de décisions ainsi que les résultats du test sont résumés dans le tableau suivant :

Base	Kstest	Résultat
$b = 2$	0.1750	on accepte H_0
$b = 7$	0.2510	on accepte H_0
$b = 17$	0.4118	on rejette H_0

TABLE III.2 – Les résultats du test de Kolmogorov-Smirnov d'uniformité pour $n = 10$.

Maintenant, générons des échantillons de taille $n = 1000$ toujours avec la suite de Halton mais en base 17, 41 et 179.

Le tableau suivant résume les résultats du test de Kolmogorov-Smirnov obtenus, tel que la valeur critique lue sur la table de Kolmogorov-Smirnov pour $n = 1000$ est $D_n = 0.0430$ puisque ($n > 35, D_n = \frac{1.36}{\sqrt{n}}$) :

Base	Kstest	Résultat
$b = 17$	0.0086	on accepte H_0
$b = 41$	0.0197	on accepte H_0
$b = 179$	0.0483	on rejette H_0

TABLE III.3 – Les résultats du test de Kolmogorov-Smirnov d'uniformité pour $n = 1000$.

III.3.2 La suite de Sobol

Nous sommes passés par les mêmes étapes qu'avec les suites de Halton, mais cette fois avec la suite de Sobol, nous avons gardé la taille qui est $n = 10$ mais nous avons utilisé une seule base c-à-d $b = 2$, le paramètre qui varie pour cette suite est la dimension(d). Les trois dimensions testées sont 5, 20 et 190. Le choix de ces dimensions est arbitraire. le tableau suivant illustre les trois suites de Sobol générées pour $n = 10$, et les dimension citées auparavant.

n	1	2	3	4	5	6	7	8	9	10
$\phi_2^5(n)$	0.5	0.25	0.75	0.625	0.125	0.875	0.375	0.8125	0.3125	0.5625
$\phi_2^{20}(n)$	0.5	0.75	0.25	0.125	0.625	0.875	0.375	0.9375	0.4375	0.1875
$\phi_2^{190}(n)$	0.5	0.75	0.25	0.125	0.625	0.875	0.375	0.4375	0.9375	0.6875

TABLE III.4 – La suite de Sobol en dimension 5, 20 et 190.

Nous allons procéder aux tests pour vérifier l'uniformité des ces suites, la statistique de décision reste la même vu qu'on n'a pas changé de taille, $D_n = 0,410$
Les statistiques de décisions ainsi que les résultats sont résumés dans le tableau suivant :

Dimension	Kstest	Résultat
$d = 5$	0.125	on accepte H_0
$d = 20$	0.10	on accepte H_0
$d = 190$	0.075	on accepte H_0

TABLE III.5 – Les résultats du test d'uniformité pour $n = 10$ de la suite de Sobol.

On remarque que ces suites préservent généralement l'uniformité de la distribution même en grandes dimension.

Pour $n = 1000$, nous avons généré avec la suite de Sobol en dimension 16,45 et 70. Le tableau suivant résume les résultats du test de Kolmogorov-Smirnov obtenus, tel que $D_n = 0,0430$.

Dimension	Kstest	Résultat
$d = 16$	0.0016	on accepte H_0
$d = 45$	0.0013	on accepte H_0
$d = 70$	0.0019	on accepte H_0

TABLE III.6 – Les résultats du test d'uniformité pour $n = 1000$ pour la suite de Sobol.

III.3.3 La suite de Faure

Idem pour cette suite la génération et les tests se feront de la même manière nous choisirons la dimension ainsi que la base comme paramètre d'entrés. Dans ce cas pour $n = 10$ on choisit $d = 3$, $b = 5$ pour le premier échantillon, le deuxième échantillon $d = 7$, $b = 11$ et pour le dernier échantillon généré $d = 21$ et $b = 23$. les trois suites sont données dans le tableau suivant :

n	1	2	3	4	5	6	7	8	9	10
$\phi_5^3(n)$	0.2	0.4	0.6	0.8	0.44	0.64	0.84	0.04	0.24	0.88
$\phi_{11}^7(n)$	0.0909	0.1818	0.2727	0.3636	0.4545	0.5455	0.6364	0.7273	0.8182	0.9091
$\phi_{23}^{21}(n)$	0.0435	0.0870	0.1304	0.1739	0.2174	0.2609	0.3043	0.3478	0.3913	0.4348

TABLE III.7 – La suite de Faure en dimension 3, 7 et 23.

En effectuant le test de kolmogorov-Smirnov tel que $D_n = 0.410$ nous avons eu les résultats résumés dans le tableau qui suit :

Dimesion	Kstest	Résultat
$d = 3$	0.12	on accepte H_0
$d = 7$	0.0909	on accepte H_0
$d = 21$	0.5652	on rejette H_0

TABLE III.8 – Les résultats du test d'uniformité pour $n = 10$ pour la suite de Faure

Les résultats du test d'uniformité de la suite de Faure en base 23, 53 et 223 pour $n = 1000$ sont présentés dans le tableau suivant :

Dimesion	Kstest	Résultat
$d = 21$	0.006	on accepte H_0
$d = 52$	0.0129	on accepte H_0
$d = 217$	0.0452	on rejette H_0

TABLE III.9 – Les résultats du test d’uniformité pour $n = 1000$ pour la suite de Faure

III.4 Génération d’échantillons exponentielle

Soit X une variable aléatoire suivant une loi exponentielle de paramètre λ ($X \rightsquigarrow \exp(\lambda)$), tel que $\lambda > 0$ et $\mathbb{E}(X) = \frac{1}{\lambda}$.

Pour simuler une variable aléatoire suivant cette loi, nous avons utilisé comme variable d’entrée les suites obtenues dans la méthode QMC.

Prenons $\lambda = 0.5$, $\mathbb{E}(X) = 2$ pour toute la suite de notre application.

III.4.1 La suite de Halton

Après avoir réaliser plusieurs tests d’uniformité sur la suite de Halton pour différentes bases supérieures à 17 lorsque $n = 10$ et supérieures à 179 quand $n = 1000$, nous avons constaté que la suite perd son uniformité.

Utilisons les suites de Halton générées auparavant pour obtenir des échantillons de la variable aléatoire X , en utilisant la formule (I.10).

n	1	2	3	4	5	6	7	8	9	10
$x_2(i)$	1.3863	2.7726	0.5754	4.1589	0.9400	1.9617	0.2671	5.5452	1.1507	2.3263
$x_7(i)$	3.8918	2.5055	1.6946	1.1192	0.6729	0.3083	7.7836	3.6248	2.3675	0.1906
$x_{17}(i)$	5.6664	4.2801	3.4692	2.8938	2.4476	2.0829	1.7746	1.5075	1.2720	1.0613

TABLE III.10 – L’échantillon de la loi exponentielle avec la suite de Halton

Nous avons procédé aux mêmes tests d’ajustement que précédemment, sauf qu’ici nous avons vérifié l’adéquation de nos échantillons obtenus par la suite de Halton à la loi exponentielle. Pour $n = 10$ la valeur critique reste la même $D_n = 0.410$.

Les statistiques du test ainsi que les résultats sont résumés dans le tableau ci-dessous :

Base	Kstest	Résultat
$b = 5$	0.0665	on accepte H_0
$b = 11$	0.0671	on accepte H_0
$b = 73$	0.4108	on rejette H_0

TABLE III.11 – Les résultats du test d'adéquation à la loi exponentielle pour $n = 10$.

Générons des échantillons de taille $n = 1000$ avec la suite de Halton en base 73, 103 et 179. Le tableau suivant résume les résultats du test de Kolmogorov-Smirnov obtenus :

Base	Kstest	Résultat
$b = 73$	0.0155	on accepte H_0
$b = 103$	0.0145	on accepte H_0
$b = 179$	0.0473	on rejette H_0

TABLE III.12 – Les résultats du test d'adéquation à la loi exponentielle pour $n = 1000$.

Récapitulatif

La encore nous avons procédé à plusieurs tests d'adéquations à la loi exponentielle pour plusieurs bases, et la conclusion est résumée dans le tableau suivant :

n	Base	Décision
10	$b < 73$	$X \rightsquigarrow \exp(1/2)$
	$b \geq 73$	X ne suit pas $\exp(1/2)$
1000	$b < 179$	$X \rightsquigarrow \exp(1/2)$
	$b \geq 179$	X ne suit pas $\exp(1/2)$

TABLE III.13 – Récapitulatif des décisions d'adéquation à la loi exponentielle.

III.4.2 La suite de Sobol

Utilisons les suites Sobol générées auparavant pour obtenir des échantillons de la variable aléatoire X suivant la loi exponentielle.

n	1	2	3	4	5	6	7	8	9	10
$x_5(i)$	1.3863	2.7726	0.5754	0.94	4.1589	0.2671	1.9617	0.4153	2.3263	1.1507
$x_{20}(i)$	1.3863	0.5754	2.7726	4.1589	0.94	0.2671	1.9617	0.1291	1.6534	3.3480
$x_{190}(i)$	1.3863	0.5754	2.7726	4.1589	0.94	0.2671	1.9617	1.6534	0.1291	0.7494

TABLE III.14 – L'échantillon de la loi exponentielle avec la suite de Sobol.

Procédons aux tests d'adéquation à la loi exponentielle, $D_n = 0.410$

Dimension	Kstest	Résultat
$d = 5$	0.15	on accepte H_0
$d = 20$	0.125	on accepte H_0
$d = 190$	0.175	on accepte H_0

TABLE III.15 – Les résultats du test d'adéquation à la loi exponentielle pour $n = 10$.

On remarque que nous acceptons tous les échantillons pour cette taille et ces dimensions.

Nous avons effectué les mêmes tests cette fois pour $n = 1000$, $D_n = 0.0430$

Dimesion	Kstest	Résultat
$d = 16$	0.0018	on accepte H_0
$d = 45$	0.002	on accepte H_0
$d = 56$	0.0022	on accepte H_0

TABLE III.16 – Les résultats du test d'adéquation à la loi exponentielle pour $n = 1000$.

Nous remarquons encore une fois que l'hypothèse que l'échantillon suit une loi exponentielle est accepté quelques soit la dimension et la taille.

III.4.3 La suite de Faure

Toujours en utilisant les échantillons obtenus de la suite de Faure auparavant, La distribution exponentielle de la suite Faure pour $n = 10$, est introduite dans le tableau suivant :

n	1	2	3	4	5	6	7	8	9	10
$x_2^3(n)$	3.2189	1.8326	1.0217	0.4463	1.6420	0.8926	0.3487	6.4378	2.8542	0.2557
$x_7^{11}(n)$	4.7958	3.4095	2.5986	2.0232	1.5769	1.2123	0.9040	0.6369	0.4013	0.1906
$x_{21}^{23}(n)$	6.2710	4.8847	4.0738	3.4984	3.0521	2.6875	2.3792	2.1121	1.8765	1.6658

TABLE III.17 – L'échantillon de la loi exponentielle avec la suite de Faure

Voici Les statistiques de décisions ainsi que les résultats du test résumés dans le tableau suivant :

Base	Dimension	Kstest	Résultat
5	3	0.1	on accepte H_0
11	7	0.09	on accepte H_0
23	21	0.4652	on rejette H_0

TABLE III.18 – Les résultats du test d'adéquation à la loi exponentielle pour $n = 10$.

Nous allons procédé aux même tests aussi pour $n = 1000$.

Base	Dimension	Kstest	Résultat
23	21	0.005	on accepte H_0
53	52	0.0119	on accepte H_0
223	217	0.0442	on rejette H_0

TABLE III.19 – Les résultats du test d'adéquation à la loi exponentielle pour $n = 1000$.

III.5 Comparaison entre les méthodes QMC, MC et RDS

Dans cette partie nous souhaitons, en premier lieu, réaliser une comparaison entre les méthodes quasi Monte Carlo, la méthode Monte carlo et la méthode de l'échantillonnage descriptif amélioré (Refined Descriptif Sampling) (RDS), en utilisant comme premier critère de comparaison l'erreur absolue et ensuite un critère plus précis comme le MSE donnés par les formules suivantes :

$$\text{Erreur absolue} = |\text{erreur}| = |\bar{X} - \mu|$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (erreur)^2$$

Notre travail consiste à calculer les différentes moyennes empirique des échantillons de la loi exponentielle en utilisant les suites générées ainsi que l'erreur absolue et le MSE.

La génération des suites de MC est obtenue par la fonction "rand" de Matlab et la génération des suites de RDS est obtenue par le générateur "getRDS".

Les résultats des deux critères sont résumés dans le tableaux suivant :

Les suites	Dim	Base	moyenne empirique \bar{X}	Erreur absolue	MSE
Halton	16	53	2.0895	0.0895	0.0123
	21	73	2.1173	0.1173	
	27	103	2.1227	0.1227	
Sobol	16	2	1.9908	0.0092	0.0001
	45	2	1.9907	0.0093	
	56	2	1.9895	0.0105	
Faure	21	23	0.4987	1.5013	2.2756
	52	53	0.4957	1.5043	
	210	211	0.4801	1.5199	
MC	\	\	1.9314	0.0686	0.0065
			1.9354	0.0646	
			1.8963	0.1037	
RDS	\	\	2.0147	0.0147	0.0058
			1.8938	0.1062	
			1.9225	0.0775	

TABLE III.20 – Calculs de la moyenne empirique, erreur absolue et (MSE) des différents échantillons générés.

Interprétation

QMC est une méthode de simulation non aléatoire qui converge plus rapidement si on la compare avec MC et RDS, nous avons réalisé 3 exécutions pour les 3 méthodes. Nous résumons l'expérience en calculant l'erreur absolue et MSE.

Comme le montre le tableau (III.20), la méthode de Quasi Monte Carlo en particulier la suite de Sobol fournit la plus faible erreur moyenne quadratique qui est égale 0,0001, pour la suite de Halton, les résultats obtenus sont plutôt bon malgré le problème que l'on peut avoir avec les dimensions élevées, quant à la suite Faure, elle nous a fournit un résultat

décevant et cela est dû à la forte corrélation entre éléments de la suite quand la dimension est élevée.

En deuxième position vient la méthode de l'échantillonnage descriptif amélioré (RDS) qui nous a donné un résultat très satisfaisant surtout pour le premier échantillon généré dont la moyenne empirique est vraiment très proche de l'espérance, enfin la méthode de Monte Carlo malgré la nature aléatoire de ces échantillons elle nous a permis d'avoir de bons résultats.

Conclusion générale

La simulation numérique est aujourd'hui un outil incontournable de compréhension du réel et d'aide à la conception. Les progrès des ordinateurs permettent la simulation réaliste et, parfois, l'optimisation de systèmes complexes[4].

la simulation numérique est utilisée dans de nombreux domaines de recherche et développement ainsi que dans le secteur financier, où elle cherche à calculer différentes approximations, cette dernière vise à sélectionner une méthode d'approximation en fonctions de critères mathématiques (convergence, estimation d'erreur), sans négliger le coût de calcul et précision numérique[9].

Ici nous nous intéressons tout particulièrement aux méthodes de Monte Carlo qui sont d'actualité car elles permettent de résoudre un problème analytique par un procédé aléatoire, seul inconvénient c'est la vitesse de convergence. Afin de remédier à ce problème les Méthodes non aléatoires font surface, la encore nous avons cités quelqu'une mais celle qui nous intéresse est la méthode quasi monte carlo dont le principal avantage est qu'elle converge plus rapidement que la méthode de Monte Carlo. Par contre, elle comporte deux inconvénients qui n'existent pas avec la méthode de Monte Carlo : la vitesse de convergence dépend de la dimension du problème (i.e. la méthode perd de son efficacité lorsque la dimension augmente) et surtout il est impossible, de produire efficacement une estimation de l'erreur d'intégration.

La plupart des travaux menés depuis les années 1990 ont eu pour objectif de proposer des solutions afin d'améliorer l'équidistribution des suites ou de proposer une mesure efficace de l'erreur commise. Depuis le début des années 2000, la solution privilégiée par les spécialistes consiste à randomiser la méthode de Quasi-Monte Carlo afin de pouvoir estimer

l'erreur d'intégration par des moyens probabilistes[45].

L'objectif ici est de montrer comment mettre en œuvre la méthode de Quasi-Monte Carlo. Nous avons introduit les outils théoriques nécessaires à la bonne compréhension des mécanismes de l'intégration Quasi Monte Carlo en insistant sur les différences fondamentales avec la méthode de Monte Carlo, notamment , nous avons étudié les suites Van Der Corput, Halton, Sobol, Faure qui font partie des familles de suites à faible discrédance[8].

Comme application nous avons procédé à la génération d'une variable aléatoire exponentielle utilisant ces suites, puis à une comparaison entre trois méthodes RDS,MC et QMC où nous avons montré que la méthode de Quasi Monte Carlo, plus particulièrement la suite de Sobol, a fourni des résultats cohérents qui confirme la théorie, la preuve de l'efficacité de cette approche par l'étude d'un problème dont la variable d'entrée est une suite à faible discrédance , en parallèle nous avons eu des résultats appréciables pour la méthode de RDS à travers laquelle nous avons obtenu une erreur minimale. Quant à la méthode MC, elle nous a fourni un résultat non négligeable.

Bibliographie

- [1] Méthodes de quasi-monte carlo et applications. http://www.irisa.fr/dionysos/pages_perso/tuffin/QMC.html.
- [2] Méthodes quasi-monte carlo (QMC). <http://www.iro.umontreal.ca/~lecuyer/ift6561/intro-p2.pdf>.
- [3] Quasi-monte carlo simulation. http://marcoagd.usuarios.rdc.puc-rio.br/quasi_mc.html#low_discrep.
- [4] Pratique de la simulation numérique. <http://www.dunod.com/sciences-techniques/sciences-techniques-industrielles/genie-mecanique/ouvrages-professionnels/pratique-de-la-simulation->, 2003.
- [5] Quasi-monte carlo et dimension effective. <http://www.math.univ-toulouse.fr/~klein/nizzan07/view.pdf>, 2003.
- [6] Matlab. <http://fr.wikipedia.org/wiki/MATLAB>, 2005.
- [7] Low discrepancy sequences. https://quanto.inria.fr/pdf_html/mc_quasi_doc/#x1-100006, 2012.
- [8] Quasi monte carlo : Halton sequence. http://www.ludovicdubrana.com/tutorial/Tutorial_12_Quasi_Monte_Carlo.pdf, 2012.
- [9] Qu'est-ce que la simulation numérique? <http://www.cea.fr/jeunes/themes/physique-chimie/la-simulation-numerique/qu-est-ce-que-la-simulation-numerique>, 2012.
- [10] The mersenne twister generator. http://en.wikipedia.org/wiki/Mersenne_twister, Mai 2013.

-
- [11] ANTONOV, I., AND SALEEV, V. An economic method of computing lp_τ -sequences. *USSR Computational Mathematics and Mathematical Physics* 19, 1 (1979), 252–256.
- [12] BARANGER, C., AND MATHIAUD, J. Méthode de monte-carlo, 2012/2013.
- [13] BENCHALAL, M., AND DJERRAH, A. Simulation des paramètres de performance d'un réseau de distribution MT par la méthode de monte-carlo. application au départ tichy. mémoire de fin d'études recherche opérationnelle, Université A.Mira Béjaia, 2000.
- [14] BIZOUARD, G. Méta-modélisation : état de l'art et application. Master's thesis, Unité de Biométrie et Intelligence Artificielle (Bia) Castanet-Tolosan, 2012.
- [15] BONTEMPI, G. Modélisation et simulation. Tech. rep., Département d'Informatique, 2008.
- [16] BOUCHARD, B. *Méthodes de Monte Carlo en Finance*. Université Paris 6, Septembre 2007.
- [17] BOUNEHAR, K., AND SMAÏLI, N. Méthodes de monte carlo par les chaines de markov (MCMC) et leurs applications. Mémoire de fin de cycle master en recherche opérationnelle, Université A.Mira Béjaia, 2011.
- [18] BRAATEN, E., AND WELLER, G. An improved low-discrepancy sequence for multi-dimensional quasi-monte carlo integration. *Journal of Computational Physics* 33, 2 (1979), 249–258.
- [19] CAFLISCH, R. Monte carlo and quasi-monte carlo methods. *Acta Numerica, Cambridge University Press* (1998), 1–49.
- [20] CHI, H., BEERLI, P., EVANS, D., AND MASCAGNI, M. On the scrambled Sobol sequence. In *Computational Science*. Springer, 2005, pp. 775–782.
- [21] DAGPUNAR, J. *Simulation and Monte Carlo : With applications in finance and MCMC*, john wiley and sons ltd ed. England, 2007.
- [22] DELYON, B. Simulation et modélisation, 2012.
- [23] DER CORPUT, J. V. Verteilungsfunktionen ii. In *Proc. Nederl. Akad. Wetensch* (1935), vol. 38, pp. 1058–1066.
- [24] FORT, G., AND MOULINES, E. Méthodes de monte carlo et applications à la finance. Tech. rep., 29 juin 2009.
- [25] FRANCO, J. *Planification d'expériences numériques en phase exploratoire pour la simulation des phénomènes complexes*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2008.

-
- [26] GAUTHIER, G. Les suites à discrédance faible, simulation monte carlo. HEC Montréal, 2009.
- [27] HALTON, J. H. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik* 2, 1 (1960), 84–90.
- [28] HAMMERSLEY, J. Monte-carlo methods for solving multivariate problems. *Annals of the New York Academy of Sciences* 86 (1960), 844–874.
- [29] HETTAK, D., AND GADOUM, L. Etude de la gestion des stocks de la matière première à l’entreprise alcost, béjaia. mémoire de fin d’études recherche opérationnelle, Université A.Mira Béjaia, 2007.
- [30] HLAJKA, E. Funktionen von beschränkter variatiou in der theorie der gleichverteilung. *Annali di Matematica Pura ed Applicata* 54, 1 (1961), 325–333.
- [31] JACQUES, J. *Contributions à l’analyse de sensibilité et à l’analyse discriminante généralisée*. PhD thesis, Université Joseph-Fourier-Grenoble I, 2005.
- [32] JOE, S., AND KUO, F. Notes on generating sobol sequences, 2008.
- [33] JOURDAN, A. *Analyse statistique et échantillonnage d’expériences simulées*. PhD thesis, Université de Pau et des pays de L’Adour, 2000.
- [34] KOCIS, L., AND WHITEN, W. Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software (TOMS)* 23, 2 (1997), 266–294.
- [35] KRYKOVA, I. *Evaluating of path-dependent securities with low discrepancy methods*. PhD thesis, Worcester Polytechnic Institute, 2003.
- [36] LAPEYRE, B., PARDOUX, E., AND SENTIS, R. *Méthodes de Monte-Carlo pour les équations de transport et de diffusion*. Springer, 1998.
- [37] LARCHER, G., AND LEOBACHER, G. Quasi-monte carlo and monte carlo methods and their application in finance. *Surveys on Mathematics for Industry* 11 (2005), 95–130.
- [38] LEI, G. Adaptive random search in quasi-monte carlo methods for global optimization. *Computers & Mathematics with Applications* 43, 6 (2002), 747–754.
- [39] MARY, J. *Étude de l’Apprentissage Actif, Application à la Conduite d’Expériences*. PhD thesis, Université Paris-Sud, 2005.
- [40] MILLET, A. *Méthodes de Monte-Carlo*. Université Paris VII et I, 2011.
- [41] OURBIH, M. Cour de simulation à évènements discrets, 2010/2011.
- [42] OURBIH, M. Cour de méthodes de monte carlo, 2012.

-
- [43] OWEN, A. Monte carlo, quasi-monte carlo, and randomized quasi-monte carlo. In *Monte Carlo and Quasi-Monte Carlo Methods* (1998), H. Niederreiter and J. Spanier, Eds., pp. 86–97.
- [44] OWEN, A. B. Latin supercube sampling for very high-dimensional simulations. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 8, 1 (1998), 71–102.
- [45] PATARD, P. *Ingénierie des produits structurés : essais sur les méthodes de simulation numérique et sur la modélisation des données de marché*. PhD thesis, Université Claude Bernard Lyon 1, 2008.
- [46] RAINVILLE, F. D., GAGNÉ, C., TEYTAUD, O., AND LAURENDEAU, D. Evolutionary optimization of low-discrepancy sequences. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 22, 2 (2012), 9.
- [47] SALAHT, F. A., AND DJABALI, Y. Stabilité forte, simulation et performance de la méthode dans un système de files d’attente M/G/1 avec service par groupe à capacité aléatoire muni de la politique (r,N). Master’s thesis, Université A.Mira de Béjaia, 2008.
- [48] SALIBY, E. Descriptive sampling : an improvement over latin hypercube sampling. In *Winter simulation Conference* (1997), pp. 230–233.
- [49] SOBOLOV, I. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki* 7, 4 (1967), 784–802.
- [50] TARI, M., AND DAHMANI, A. Refined descriptive sampling : A better approach to monte carlo simulation. *Simulation Modelling Practice and Theory* 14, 2 (2006), 143–160.
- [51] TRIGNAN, J. *Probabilités et statistiques et leurs applications : Brevet de technicien supérieur*, bréal ed. instituts universitaires de technologie, octobre 1990.
- [52] TUFFIN, B. Variance reduction order using good lattice points in monte carlo methods. *Computing* 61, 4 (1998), 371–378.
- [53] TUFFIN, B. *Modélisation mathématique pour la conception, l’analyse quantitative et le contrôle de la qualité de service des systèmes*. PhD thesis, Université de Rennes 1, 2006.
- [54] TUFFIN, B., AND RUBINO, G. *Simulations et méthodes de Monte Carlo*, techniques ingénieur ed. Rennes, October 2007.
- [55] VARET, S. *Développement de méthodes statistiques pour la prédiction d’un gabarit de signature infrarouge*. PhD thesis, Université Toulouse III - Paul Sabatier, 2010.

- [56] WANG, X. A new measure of irregularity of distribution and quasi-monte carlo methods for global optimization. *Computers & Mathematics with Applications* 43, 6 (2002), 657–669.
- [57] WANG, Y., AND HICKERNELL, F. *An historical overview of lattice point sets*. Springer, 2002.
- [58] WONG, T., LUK, W., AND HENG, P. Sampling with hammersley and halton points. *Journal of graphics tools* 2, 2 (1997), 9–24.

Annexes

$$D = \sup |F_n(x) - F(x)|$$

n / α	0.10	0.05	0.01	n / α	0.10	0.05	0.01
1	0.950	0.875	0.995	14	0.314	0.349	0.418
2	0.776	0.842	0.929	15	0.304	0.338	0.404
3	0.642	0.708	0.828	16	0.295	0.328	0.392
4	0.564	0.624	0.733	17	0.286	0.318	0.381
5	0.510	0.565	0.669	18	0.278	0.309	0.371
6	0.470	0.521	0.618	19	0.272	0.301	0.363
7	0.438	0.486	0.577	20	0.264	0.294	0.356
8	0.411	0.457	0.543	25	0.240	0.270	0.320
9	0.388	0.432	0.514	30	0.220	0.240	0.290
10	0.368	0.410	0.490	35	0.210	0.230	0.270
11	0.352	0.391	0.468	>35	$1.22/n^{1/2}$	$1.36/n^{1/2}$	$1.63n^{1/2}$
12	0.338	0.375	0.450				
13	0.325	0.361	1.433				

n : l'effectif total, α : le risque d'erreur.

FIGURE III.1 – Table de Kolmogorov-Smirnov

TABLE DU CHI-DEUX : $\chi^2(n)$



n P	0.90	0.80	0.70	0.50	0.30	0.20	0.10	0.05	0.02	0.01
1	0,0158	0,0642	0,148	0,455	1,074	1,642	2,706	3,841	5,412	6,635
2	0,211	0,446	0,713	1,386	2,408	3,219	4,605	5,991	7,824	9,210
3	0,584	1,005	1,424	2,366	3,665	4,642	6,251	7,815	9,837	11,341
4	1,064	1,649	2,195	3,357	4,878	5,989	7,779	9,488	11,668	13,277
5	1,610	2,343	3,000	4,351	6,064	7,289	9,236	11,070	13,388	15,086
6	2,204	3,070	3,828	5,348	7,231	8,558	10,645	12,592	15,033	16,812
7	2,833	3,822	4,671	6,346	8,383	9,803	12,017	14,067	16,622	18,475
8	3,490	4,594	5,527	7,344	9,524	11,030	13,362	15,507	18,168	20,090
9	4,168	5,380	6,393	8,343	10,656	12,242	14,684	16,919	19,679	21,666
10	4,865	6,179	7,267	9,342	11,781	13,442	15,987	18,307	21,161	23,209
11	5,578	6,989	8,148	10,341	12,899	14,631	17,275	19,675	22,618	24,725
12	6,304	7,807	9,034	11,340	14,011	15,812	18,549	21,026	24,054	26,217
13	7,042	8,634	9,926	12,340	15,119	16,985	19,812	22,362	25,472	27,688
14	7,790	9,467	10,821	13,339	16,222	18,151	21,064	23,685	26,873	29,141
15	8,547	10,307	11,721	14,339	17,322	19,311	22,307	24,996	28,259	30,578
16	9,312	11,152	12,624	15,338	18,418	20,465	23,542	26,296	29,633	32,000
17	10,085	12,002	13,531	16,338	19,511	21,615	24,769	27,587	30,995	33,409
18	10,865	12,857	14,440	17,338	20,601	22,760	25,989	28,869	32,346	34,805
19	11,651	13,716	15,352	18,338	21,689	23,900	27,204	30,144	33,687	36,191
20	12,443	14,578	16,266	19,337	22,775	25,038	28,412	31,410	35,020	37,566
21	13,240	15,445	17,182	20,337	23,858	26,171	29,615	32,671	36,343	38,932
22	14,041	16,314	18,101	21,337	24,939	27,301	30,813	33,924	37,659	40,289
23	14,848	17,187	19,021	22,337	26,018	28,429	32,007	35,172	38,968	41,638
24	15,659	18,062	19,943	23,337	27,096	29,553	33,196	36,415	40,270	42,980
25	16,473	18,940	20,867	24,337	28,172	30,675	34,382	37,652	41,566	44,314
26	17,292	19,820	21,792	25,336	29,246	31,795	35,563	38,885	42,856	45,642
27	18,114	20,703	22,719	26,336	30,319	32,912	36,741	40,113	44,140	46,963
28	18,939	21,588	23,647	27,336	31,391	34,027	37,916	41,337	45,419	48,278
29	19,768	22,475	24,577	28,336	32,461	35,139	39,087	42,557	46,693	49,588
30	20,599	23,364	25,508	29,336	33,530	36,250	40,256	43,773	47,962	50,892

Pour $n > 30$, on peut admettre que $\sqrt{2\chi^2} - \sqrt{2n-1} \approx N(0,1)$

FIGURE III.2 – Table de Khi-deux

Résumé

La méthode de Monte Carlo est l'une des méthodes numériques les plus polyvalentes et les plus utilisées. Son taux de convergence étant de $o(\frac{1}{\sqrt{n}})$, est indépendant de la dimension, ce qui montre que Monte Carlo peut être très robuste, mais aussi lente. Ici nous présentons une introduction aux méthodes de Monte Carlo pour les problèmes d'intégration, y compris les méthodes d'échantillonnages et techniques de réduction de la variance. L'accélération de la convergence pour la méthode de Monte Carlo est atteinte en utilisant des suites à faible discrédance, qui sont une alternative déterministe à des suites aléatoires ou pseudo-aléatoire. La méthode résultante, est appelée Quasi-Monte Carlo.

Les méthodes de Quasi-Monte Carlo sont basées sur l'idée que les techniques de Monte Carlo aléatoires peuvent souvent être améliorées par le remplacement de la source de nombres aléatoires avec une suite déterministe distribuée plus uniformément. Le taux de convergence est de l'ordre de $o((\log n)^d n^{-1})$, la encore nous présentons une méthode quasi-Monte Carlo pour l'intégration. Toutes les méthodes sont réalisées en utilisant des nombres pseudo-aléatoires, et des suites à faible discrédance : Van Der Corput, Halton, Hammersley, Sobol, et Faure. Les résultats numériques confirment l'amélioration de la convergence de la méthode proposée.

Mots clés : Monte Carlo, Quasi Monte Carlo, Convergence, suite à faible discrédance.

Abstract

Monte Carlo is one of the most versatile and widely used numerical methods. Its convergence rate, $o(\frac{1}{\sqrt{n}})$, is independent of dimension, which shows Monte Carlo to be very robust but also slow. Here we present an introduction to Monte Carlo methods for integration problems, including sampling methods and variance reduction techniques. Accelerated convergence for Monte Carlo method is attained using low discrepancy sequences, which are a deterministic alternative to random or pseudo-random sequences. The resulting method, called Quasi-Monte Carlo.

Quasi-Monte Carlo methods are based on the idea that random Monte Carlo techniques can often be improved by replacing the source of random numbers with a more uniformly distributed deterministic sequence. The convergence rate of approximately $o((\log n)^d n^{-1})$. Here again, we present a quasi-Monte Carlo method for integration. All methods are realized using pseudo random numbers and low discrepancy sequences : Van Der Corput, Halton, Hammersley, Sobol, et Faure. The numerical results confirm the improved convergence of the proposed method.

Keywords : Monte Carlo, Quasi Monte Carlo, Convergence, Low discrepancy sequences.