

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A.MIRA-BEJAIA

Faculté des Sciences Exactes
Département de mathématiques



Mémoire

Présentée par :

BOUBALOU Meriem

&

OUKACHBI Amina

Pour l'obtention du diplôme de

Master

Filière : Mathématiques

Option : Statistique et Analyse Décisionnelle

Thème

**La comparaison entre les méthodes
hypercube latin et descriptif amélioré à
travers un modèle de file d'attente M/G/1**

Devant le jury composé de :

M ^{me} L. BOURAINE	Présidente	M.C.A	Université de Bejaia
M ^{me} S. GUEBLI	Examinatrice	M.C.B	Université de Bejaia
M ^{me} M. OURBIH	Rapporteur	Professeur	Université de Bejaia

Promotion 2015

Remerciements

Nous tenons à remercier d'abord Dieu qui nous a procuré du courage et de la volonté pour achever ce travail et aussi de nous avoir donné la force d'aller au bout de notre objectif.

Un grand merci à Madame O'URBIH notre promotrice à qui nous voudrions lui témoigner notre gratitude pour sa patience et son soutien qui nous a été précieux durant toute la période de ce travail.

Nous remercions également M^{me} Bouraine et M^{me} Guebli qui ont accepté d'être membres de jury .

Merci à toute personne qui a contribué à la réalisation de ce mémoire.

Dédicaces

Je dédie ce modeste travail :

A mes très chers parents. Tous les mots du monde ne sauraient exprimer l'immense amour que je leurs porte, ni la profonde gratitude que je leurs témoigne pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour mon éducation et mon bien-être, j'espère avoir répondu aux espoirs que vous avez fondés en moi.

A mes chères sœurs Nadjet, Halima et Manel

A mon cher frère Lotfi

A tous les membres de ma famille grands et petits

A mes chers amis Bouchra, Hanna, Khaled, Sabiha et Meriem

A ma très chère binôme et amie qui m'a vraiment aidé à réaliser ce travail, et à toute sa famille.

A ceux qui m'ont aidé de près ou de loin dans la réalisation de ce travail

Amina

Dédicaces

Je dédie ce modeste travail :

A mes très chers parents. Tous les mots du monde ne sauraient exprimer l'immense amour que je leurs porte, ni la profonde gratitude que je leurs témoigne pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour mon éducation et mon bien-être, j'espère avoir répondu aux espoirs que vous avez fondés en moi.

A mes chères sœurs

A mes chers frères

A ma chère grand mère

A mes chères tantes

A tous les membres de ma famille grands et petits

A ma très chère binôme et amie qui m'a vraiment aidé à réaliser ce travail, et à toute sa famille.

A ceux qui m'ont aidé de près ou de loin dans la réalisation de ce travail

Meriem

Table des matières

Introduction général	7
Chapitre 1: Généralités sur l'échantillonnage et la simulation	9
1.1. Introduction	10
1.2. Notion d'échantillonnage	10
1.2.1. Population et échantillon	10
1.2.2. L'échantillonnage	10
1.2.3. Avantage de l'échantillonnage	10
1.2.4. Méthode d'échantillonnage	10
1.3. Simulation	12
1.3.1. Définition de la simulation.....	12
1.3.2. Les étapes d'une étude de simulation	12
1.3.3. Description de l'organigramme	15
1.3.4. Génération des nombres aléatoires	15
1.3.5. Générateur des nombres pseudo-aléatoires	16
1.3.6. Les générateur congruentielle	17
1.3.7. Qualité d'un bon générateur	18
1.4. Simulation de variable aléatoires suivant une loi donnée	18
1.4.1. Méthode d'inversion	18
1.4.2. Méthode de rejet	20
1.5. La simulation à événements discrets	20
1.5.1. Définition	20
1.5.2. Exemple de simulation	21
1.6. Simulation de Monte Carlo.....	22

1.6.1. La méthode de Monte Carlo	23
1.6.1.1. Présentation de la méthode	23
1.6.1.2. Vitesse de convergence	23
1.6.1.3. Calcul de la variance.....	24
1.6.1.4. L'approximation des intégrales.....	24
1.6.1.5. Exemple	25
1.6.1.6. Avantage de la méthode MC	25
1.7. L'échantillonnage descriptif	25
1.8. Conclusion	27
Chapitre 2: Echantillonnage descriptif amélioré Et hypercube latin	28
2.1. Introduction	29
2.2. Echantillonnage descriptif amélioré	29
2.2.1. Principe d'EDA	29
2.2.2. Générer des valeurs avec EDA	30
2.2.3. Algorithme d'EDA	30
2.2.4. La différence entre EA, ED et EDA	31
2.2.5. Exemple	32
2.3. Echantillonnage hypercube latin	33
2.3.1. Méthode Hypercube Latin	33
2.3.2. Quelques définitions sur LHD	33
2.3.3. Principe de la méthode	34
2.3.4. Avantages	37
2.3.5. Inconvénients	37
2.4. Conclusion	37
Chapitre 3: Les systèmes de files d'attente (M/M/1, M/G/1).....	38
3.1. Introduction	39

3.2. Classification des systèmes d'attente	39
3.3. Intervalle entre deux événements	40
3.4. Caractéristiques d'un système d'attente	41
3.4.1. Définitions, formules de Little	41
3.5. Système de file d'attente M/M/1	41
3.5.1. Régime stationnaire du système	42
3.5.2. Régime stationnaire	43
3.5.3. Caractéristiques du système M / M / 1	44
3.6. Système d'attente M/G/1.....	45
3.6.1. Chaîne de Markov induite	45
3.6.2. Régime stationnaire du système	42
3.6.3. Caractéristiques du système M/G/1	48
3.7. Conclusion	49
Chapitre 4: Application et comparaison	50
4.1. Introduction	51
4.2. Présentation de MATLAB.....	51
4.3. Comment générer des variables avec la méthode LHS	52
4.4. Comment générer des variables avec la méthode EDA	53
4.5. Problématique	53
4.6. Résolution avec la simulation	54
4.7. Résultats de simulation pour différentes lois	55
4.7.1. En utilisant le critère de la variance	55
4.7.2. En utilisant le critère de la valeur absolue	64
Conclusion général	66
Bibliographie	68
Annexe.....	70

Liste des figures

1.1 Présentation du modèle simulation 12

1.2. Etapes de la simulations 14

2.1 .Illustration de l'échantillon EDA de taille n 26

2.2. Illustration d'une histoire EDA 26

2.3. Hypercube latin de taille 10 pour deux variables d'entrée X_1 et X_2 32

2.4 . Exemple d'échantillonnage LHS pour 6 points en dimension 2..... 33

4.1: File d'attente simple à un serveur 50

Liste des Tableaux

1.1. Résultats obtenues de la méthode ED	23
1.2. Résultats obtenues par la méthode ED	24
2.1: Illustration de la méthode EDA	29
2.2. La moyenne observée d'une distribution uniforme en utilisant trois sous-ensemble de nombre premier.....	29
2.3. Plan en hypercube latin LHD(10,2) (transposé).....	31
4.1. La moyenne des paramètres estimer par la simulation en utilisant LHS	55
4.2. La moyenne des paramètres estimer par la simulation en utilisant EDA	56
4.3. La variance des estimateurs obtenue par LHS où $\rho = 0.3$	56
4.4. La variance des estimateurs obtenue par EDA où $\rho = 0.3$	57
4.5. La moyenne des paramètres estimer par la simulation en utilisant LHS où $\rho = 0.9$	57
4.6. La moyenne des paramètres estimer par la simulation en utilisant EDA où $\rho = 0.9$	58
4.7. La variance des estimateurs obtenue par LHS où $\rho = 0.9$	58
4.8. La variance des estimateurs obtenue par EDA où $\rho = 0.9$	59
4.9. Les résultats de simulation d'une file M/G/1 par LHS	60
4.10. Les résultats de simulation d'une file M/G/1 par EDA	60

4.11. La variance des estimateurs obtenue par LHS où $\rho = 0.3$	61
4.12. La variance des estimateurs obtenue par EDA où $\rho = 0.3$	61
4.13. Les résultats de simulation d'une file M/G/1 par LHS où $\rho = 0.9$	62
4.14. Les résultats de simulation d'une file M/G/1 par EDA où $\rho = 0.9$	62
4.15. La variance des paramètres obtenue par LHS où $\rho = 0.9$	63
4.16. La variance des paramètres obtenue par LHS où $\rho = 0.9$	63
4.17. Les valeurs empiriques des erreurs absolues obtenues par EDA	64
4.18. Les valeurs empiriques des erreurs absolues obtenue par LHS	65

Introduction générale

De manière générale, la simulation permet d'étudier et expérimenter un système donné dont on connaît les interactions complexes, de mesurer les effets de certains changements dans les interactions sur le comportement du système, d'expérimenter de nouvelles situations. Remarquons de plus que si l'on cherche une représentation fidèle des phénomènes observés, on est rapidement confronté à des difficultés dues aux calculs non explicites. Les techniques de simulation vont nous permettre d'approcher numériquement ces calculs.

Effectuer une simulation consiste à générer des données qui sont des réalisations de variables déterministes ou aléatoires selon des lois données, afin d'étudier et de comprendre le fonctionnement de systèmes industriels, scientifiques, politiques, etc.

Le but de la statistique est de dégager les significations de données, numériques ou non, obtenues au cours de l'étude d'un phénomène. Il faut distinguer les données statistiques qui sont les résultats d'observations recueillies lors de l'étude d'un phénomène, et la méthode statistique qui a pour objet l'étude rationnelle des données [6].

La méthode statistique comporte plusieurs étapes dont notre cas nous allons voir la statistique mathématique ou inductive, alors cette dernière c'est l'ensemble des méthodes qui permettent de faire des prévisions, des interpolations sur une population à partir des résultats recueillis sur un échantillon[6].

Cette statistique utilise des repères de référence qui sont les modèles théoriques (lois de probabilités). Elle nécessite la recherche d'échantillons qui représentent le mieux possible la diversité de la population entière. L'étude sur échantillon se justifie pour réduire le coût élevé et limiter la destruction d'individus pour obtenir la réponse statistique [6]. De nombreuses méthodes d'échantillonnage existent la méthode Monte Carlo, hypercube latin (LHS), Quasi Monte Carlo, Quasi Monte Carlo randomisées, l'échantillonnage Descriptif (ED), l'échantillonnage Supercube Latin et l'échantillonnage Descriptif amélioré (EDA).

Par ailleurs, la théorie des files d'attente est une technique de la Recherche opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer et de déterminer ses caractéristiques pour aider les gestionnaires dans leurs prises de décisions.

Le but de notre travail est d'appliquer les deux méthodes LHS et EDA pour étudier les propriétés monotonicité du modèle $M/G/1$ et comparer les résultats obtenues.

Ce mémoire est constitué d'une introduction générale, de quatre chapitres, d'une conclusion générale et d'une bibliographie.

Le premier chapitre comprend les rappels et quelques théorèmes élémentaires de la théorie générale de statistique ainsi dans le second chapitre, on donne une explication détaillée sur les deux méthodes LHS et EDA, Le troisième chapitre est consacré aux concepts et techniques de base de la théorie des files d'attente classiques.

Dans le quatrième chapitre, nous réalisons une application pour pouvoir faire une comparaison entre LHS et EDA. Pour clore ce travail une conclusion générale sera présentée.

Chapitre - I

Généralités sur l'échantillonnage et
la simulation

1.1. Introduction

Certains paramètres d'entrée d'une simulation ne sont pas connus exactement, plusieurs méthodes sont envisagées.

Les méthodes d'échantillonnage peuvent nous donner la possibilité de connaître ces inconnus.

Les techniques d'échantillonnage étant parfois pareille et des fois le contraire, dans ce qui suit nous répertorierons les différentes particularités de ces techniques et nous parlerons de la simulation.

1.2. Notion d'échantillonnage

1.2.1. Population et échantillon

On appelle population la totalité des unités de n'importe quel genre prises en considération par le statisticien. Elle peut être finie ou infinie.

Un échantillon est un sous-ensemble de la population étudiée.

1.2.2. L'échantillonnage

L'échantillonnage représente l'ensemble des opérations qui ont pour objet de prélever un certain nombre d'individus dans une population donnée.

1.2.3. Avantages de l'échantillonnage

- ❖ Coût moindre.
- ❖ Gain de temps.

1.2.4. Méthodes d'échantillonnage

a. L'échantillonnage aléatoire (EA):

Dans cet échantillonnage chaque élément de la population a une probabilité d'être choisi.

❖ L'échantillonnage stratifié :

Il s'agit de subdiviser une population hétérogène en strates (sous-groupe). Cette méthode consiste à retrouver dans l'échantillon les mêmes proportions pour chacune des strates selon les caractéristiques choisies pour l'étude dans la population visée.

Exemple:

On a une population de 200 individus.

Sexe	Nombre d'individu
Hommes	84
Femmes	116
Total	200

Hommes: 84 ==> 42% de la population

Femmes: 116 ==> 58% de la population

84 + 116 = 200 individus de ma population

On veut un échantillon de taille 50 et on veut qu'il représente fidèlement notre population.

Donc on utilise les proportions pour obtenir un échantillon représentatif.

Hommes: 50 * 42% = 21

Femmes: 50 * 58% = 29

21 + 29 = 50 individus de mon échantillon

Sexe	Nombre d'individu	Échantillon
Hommes	84	21
Femmes	116	29
Total	200	50

❖ **L'échantillonnage aléatoire simple :**

Tous les individus ont la même probabilité de faire partie de l'échantillon.

b. L'échantillonnage non aléatoire :

Chaque élément de la population n'a pas une chance égale d'être choisi. Cet échantillon risque d'être moins représentatif de la population.

❖ **L'échantillonnage au jugé :**

On utilise la méthode d'échantillonnage au jugé lorsqu'on prélève un échantillon en se fondant sur certains jugements au sujet de l'ensemble de la population.

1.3. Simulation

1.3.1. 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'activité, 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 [14].

Dans une étude de simulation, un modèle logique est construit et utilisé comme véhicule pour l'expérimentation. Le modèle est illustré à la figure 1. Les distributions des variables d'entrée sont supposés connues, tandis que les distributions des variables de sortie sont inconnus. Lorsque le problème est simulé, les variables aléatoires d'entrée sont remplacés par des échantillons. En conséquence, les variables de sortie sont également remplacés par des échantillons.

Par conséquent, des expériences sont réalisées sur le modèle et les paramètres inconnus de variables aléatoires de sortie d'intérêt sont estimées [9].

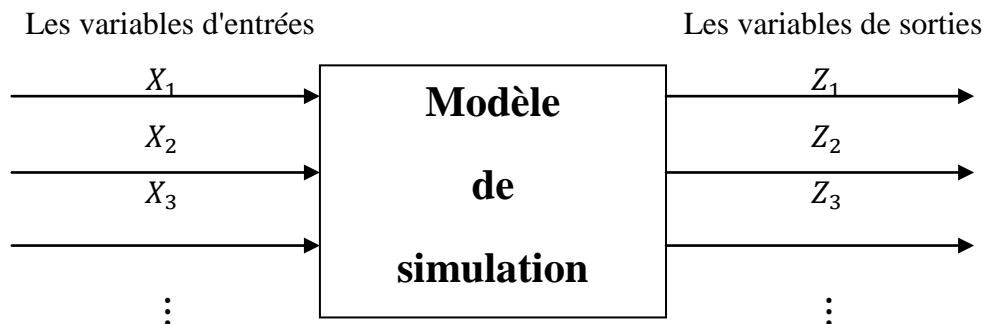


Figure 1.1 Présentation du modèle de simulation : un ensemble de variables aléatoires d'entrée est transformé en un ensemble de variables aléatoires de sortie.

1.3.2. Les étapes d'une étude de simulation

La simulation, comme les autres techniques d'évaluation des performances, est souvent perçue comme une technique qui tient "autant de l'art que de la science". De ce point de vue, le succès du procédé est largement lié à l'expertise du responsable de l'expérience. Cela dit, on constate l'existence d'une communauté fertile de spécialistes appliqués à faire évoluer la discipline. Un résultat immédiat de cette activité est le développement de méthodologies qui guident l'analyste en structurant les éléments dont on doit tenir compte dans une expérience de simulation. Les étapes de la méthodologie ne doivent pas être suivies

d'une façon rigide et séquentielle, mais comme un processus itératif qui s'adapte selon le type et l'évolution du système à étudier. Comme la simulation peut être utilisée pour faire des ajustements au niveau du modèle et du programme si c'est nécessaire alors la figure au-dessous décrit les différentes étapes de simulation [11].

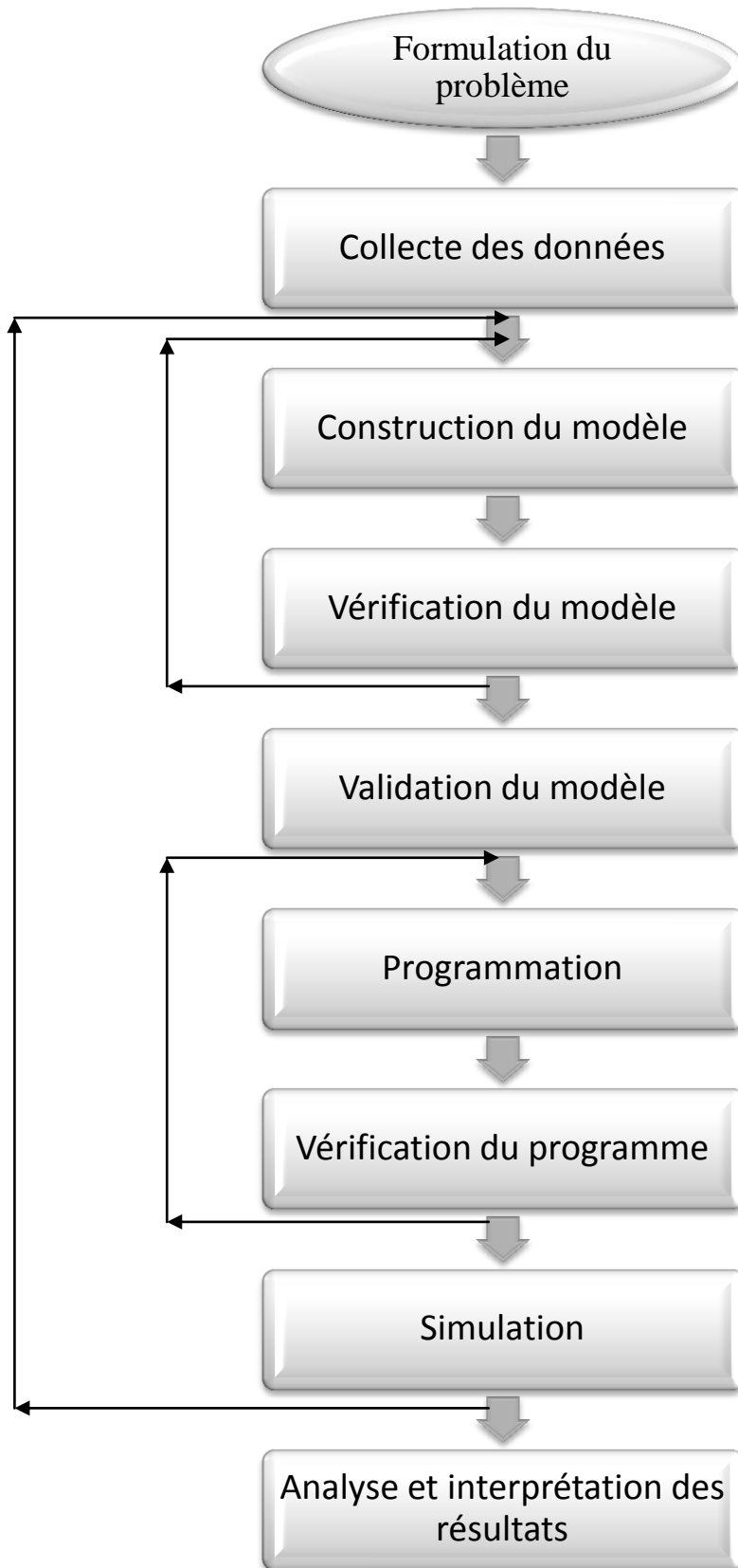


Figure 1.2. Etapes de la simulations

1.3.3. Description de l'organigramme

a) Formulation du problème

Le concepteur se doit d'isoler et de comprendre les caractéristiques essentielles du problème. Il s'agit de déterminer la frontière entre le système et son environnement, élaborer les objectifs de l'organisation, identifier les contraintes sur les activités possibles et établir les liens entre les entités du système et son environnement.

b) Construction du modèle

Le concepteur se doit de refléter le système réel avec le plus de fidélité, c'est à dire, capter l'essence du problème en gardant le modèle à sa plus simple expression.

Lors de la construction du modèle, il faut éviter deux principaux dangers : premièrement s'empêtrer dans les détails et deuxièmement rendre le phénomène grossier. Il existe deux manières différentes d'aboutir à un modèle fidèle au système réel :

- Démarrer d'un modèle simplifié pour l'améliorer au fur et à mesure jusqu'à aboutir au modèle final qui représentera au mieux le système étudié.
- Segmenter le système en sous-systèmes, les modéliser indépendamment pour les combinés afin d'aboutir ainsi au modèle final.

Le modèle se doit d'être le plus représentatif possible des paramètres sur les quels porte l'étude.

c) Collecte des données

Cette étape consiste à procéder à une identification précise des données nécessaires, puis de les collecter et de les ajuster par des distributions de probabilités connues.

Le recueil des données doit être exécuté consciencieusement puisqu'il est impossible d'obtenir des résultats satisfaisants à partir de données erronées.

d) Vérification du modèle

Il s'agit d'évaluer les performances de notre programme en vérifiant si sa transcription informatique correspond bien au modèle. Le concepteur doit vérifier si la logique du système est conservée.

e) Validation du modèle

Cette étape consiste à tester le degré de correspondance entre les résultats du système réel et ceux du modèle de simulation. Il y a plusieurs aspects dans cette validation:

- Validation du modèle écrit, sous les deux aspects de conformité du modèle écrit au modèle conçu, et de validation du produit logiciel.
- Validation des données fournies (paramètres numériques, types des lois, etc.).
- Validation des résultats (analyse statistique des résultats, pour fournir un intervalle de confiance).

f) Programmation

Cette étape consiste à d'écrire le programme de notre modèle.

g) Vérification du programme

Cette étape consiste à vérifier si le programme reflète la réalité de notre modèle.

h) Exécution de la simulation

Cette étape consiste à effectuer plusieurs exécutions et de recueillir les résultats obtenus. Durant ces exécutions, on peut réviser le modèle et le programme si cela s'avère nécessaire.

i) Analyse et interprétation des résultats

Une fois la fiabilité des résultats prouvée, le concepteur pourra les interpréter et faire les recommandations aux décideurs.

1.3.4. Génération des nombres aléatoires

Les générateurs de nombres aléatoires sont des dispositifs capables de produire une séquence de nombres dont on ne peut pas tirer facilement des propriétés déterministes.

Il existe depuis très longtemps des méthodes pour obtenir de tels nombres, notamment utilisées dans les jeux de hasard, comme par exemple le lancement de dés ou le jeu de la roulette [11].

1.3.5. Générateur des nombres pseudo-aléatoires

On entend par suite de nombres pseudo-aléatoires, toute suite constructible dont les propriétés statistiques sont proches de celles de la suite $(U_1; U_2; \dots)$ où les U_i sont des variables aléatoires indépendantes et de loi uniforme sur $[0;1]$: La génération de suite de nombres pseudo-aléatoires est une tâche complexe. Tous les générateurs pseudo-aléatoires sont basés sur des générateurs uniformes, c'est-à-dire des générateurs dans lesquels tous les nombres qui peuvent être produits ont une égale fréquence d'apparition.

Les générateurs les plus utilisés sont des générateurs récurrents: l'obtention d'un nombre pseudo-aléatoire est fondée sur un ou plusieurs nombres préalablement obtenus. Parmi les algorithmes récurrents, le plus classique, inventé par Lehmer, s'appelle méthode congruentielle multiplicative [11].

1.3.6. Le générateur congruentielle

Les générateurs dits congruentiels ont été inventés en 1948 par D.H Lehmer. Ce sont des générateurs pseudo-aléatoires efficaces, simples et présentant de nombreuses qualités. On peut mener une étude mathématique assez complète de leurs propriétés ce qui est une garantie pour la qualité du générateur.

Un générateur congruentiel est une suite de nombres entiers définie par un premier terme X_0 arbitraire, notre germe, et par la relation de récurrence [5]:

$$\left\{ \begin{array}{l} X_{n+1} = aX_n + c(\text{mod } m) \quad \text{telle que } n \geq 0 \\ X_0 \text{ donné} \end{array} \right.$$

Soient a, c, m et X_0 sont des entiers telle que:

m : modulo $m > 0$

a : multiplicateur $0 \leq a < m$

c : l'incrément $0 \leq c < m$

X_0 = la valeur de départ $0 \leq X_0 < m$

- ❖ Dans le cas ou $c=0$ on obtient la méthode congruentielle multiplicative ou linéaire.
- ❖ Le cas ou $c \neq 0$ on obtient la méthode congruentielle mixte[11].

1.3.7. Qualités d'un bon générateur

Qu'attend-on réellement d'un générateur de nombres pseudo-aléatoires? La réponse à cette question conditionne la méthode à choisir, car dans ce domaine comme ailleurs, tout est affaire de compromis. En général on privilégie deux critères:

- la vitesse: il faut que le calcul du nombre pseudo-aléatoire suivant soit rapide. Il n'est en effet pas rare de devoir générer des millions de nombres.
- la simplicité: une méthode très compliquée sera très difficile à programmer et surtout à tester. En général, on dit même en forme de boutade que plus l'algorithme est compliqué moins la séquence sera aléatoire [11].

1.4. Simulation de variable aléatoires suivant une loi donnée

1.4.1. Méthode d'inversion

La méthode d'inversion appelé aussi méthode de transformation n'est utilisée que si la fonction de densité est connu analytiquement et peut être intégré facilement [11].

Pour être plus précis, nous traitent comme exemple le cas de la loi exponentielle et le cas de loi uniforme

❖ La loi exponentielle

La loi exponentielle de paramètre λ a pour fonction de répartition F

$$F_X(x) = \begin{cases} 1 - e^{-\lambda x} & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$$

L'inverse de cette fonction est facilement calculable, $U \in]0,1[$

$$\begin{aligned} F_X(x) = u &\Leftrightarrow e^{-\lambda x} = (1 - u) \\ &\Leftrightarrow x = F^{-1}(u) = -\frac{1}{\lambda} \log(1 - u) \end{aligned}$$

On a $1 - e^{-\lambda x} \sim U[0,1]$ alors $x = -\frac{1}{\lambda} \ln(u)$, donc pour simuler une v.a exponentielle de paramètre λ , il suffit de générer des nombres aléatoire U_i de v.a uniforme sur $[0,1]$ et puis en déduit les réalisations x_i telle que $x_i = -\frac{1}{\lambda} \ln(u_i)$ [11].

❖ **Loi Gamma**

$X \sim$ loi d'Erlang (n, λ) ou gamma (n, λ) $\lambda > 0$.

Soit Y une variable aléatoire \sim exponentielle (λ) .

On sait que la somme de n variables aléatoires indépendantes de même loi que Y suit la loi gamma (n, λ) .

La simulation de la loi exponentielle obtenue par la méthode d'inversion est :

$$y_i = -\frac{1}{\lambda} \ln(u_i) \text{ tel que } U_i \sim U[0,1].$$

$$\text{Puisque } X = \sum_{i=1}^p y_i \text{ alors } x_i = \sum_{i=1}^p \left(-\frac{1}{\lambda}\right) \ln(u_i) = \left(-\frac{1}{\lambda}\right) \ln\left(\prod_{i=1}^p u_i\right)$$

La simulation de la loi gamma (p, λ) par la méthode d'inversion est :

$$x_i = \left(-\frac{1}{\lambda}\right) \ln\left(\prod_{i=1}^p u_i\right) \text{ tel que } U_i \sim U[0,1].$$

Exemple :

Générer deux observation de lois d'Erlang(2), elles sont obtenue par $x_i = -\frac{1}{\lambda} \ln\left(\prod_{i=1}^2 u_i\right)$

$$u_i = 0.44 - 0.58 - 0.11 - 0.87$$

$$x_1 = -2 \ln\left(\prod_{i=1}^2 u_i\right) = -2 \ln(0.44 \times 0.58) = 2.73$$

$$x_2 = -2 \ln(0.11 \times 0.87) = 4.69$$

❖ **Loi uniforme**

La loi uniforme $[a, b]$ sur (a, b) a pour fonction de répartition F

$$F_X(x) = \begin{cases} 0 & \text{si } x < a \\ \frac{x-a}{b-a} & \text{si } a \leq x < b \\ 1 & \text{si } b \leq x \end{cases}$$

L'inverse de cette fonction est facilement calculable, $u \in]0,1[$

$$\begin{aligned} F(x) = u &\Leftrightarrow u = \frac{x-a}{b-a} \\ &\Leftrightarrow x = F^{-1}(u) = (b-a)u + a. \end{aligned}$$

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

$$x = (b - a)u + a.$$

1.4.2. Méthode de rejet

Si l'on désire générer des valeurs d'une variable aléatoire avec une 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 U domaine identique à celui de f et une constante c positive.

- ❖ Normaliser le domaine de $f(x)$ à l'aide d'un échelle c , de sorte que:

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

- ❖ Définir X comme fonction linéaire de u :

$$x = a + (b - a)u$$

- ❖ Générer une paire de nombres aléatoires (u_1, u_2)
- ❖ Chaque fois que l'on rencontre une paire de nombres aléatoires satisfaction:

$$u_2 \leq c[f(a + (b - a)u_1)]$$

On accepte $X = a + (b - a)u_1$ comme variable aléatoire suivant $f(x)$ [10].

1.5. La simulation à événements discrets

- **Activité**

C'est un processus qui provoque un changement d'état dans un système est généralement appelle un événement.

On appelle activité **B** (Bound) des activités qui sont exécutées à l'instant où elles sont prévus.

On appelle activité **C** (conditionnel) des activités qui se produisent lorsque certaines conditions sont réunies

1.5.1. Définition

La simulation à événements discrets [11] est la simulation d'un système dont l'état ne peut changer que lors d'instantes temporelles distincts.

En termes mathématiques nous pouvons dire que l'ensemble d'instantes temporelles aux quels le système peut changer est dénombrable.

Il existe 4 méthodes dans la simulation à événements discrets :

1. Approche des 3 phases.
2. Approche orientée activités.
3. Approche orientée évènement.
4. Approche par interaction des processus.

1. Approche des 3 phases

Cette méthode est présente par Tocher est décrite de la manière suivante :

Phase **A** : déterminer quand l'évènement suivant se produit et avancé le temps simule à ce point.

Phase **B** : exécuté toute les activités **B** qui doivent se produire à cet instant.

Phase **C** : exécuté tous les activités **C**.

D'où les conditions sont maintenant satisfaites dans cette méthode le temps simule est contrôler par la technique du prochaine évènement appelé méthode à pour variable

Le temps avance lorsqu'un évènement se produit pouvant un changement d'état d'un système.

2. Approche orientée activités

Lorsqu'on utilise cette approche, l'horloge est avancée par intervalle de temps discret uniforme et le système est examiné à chaque une des unités de temps de l'horloge pour voir c'est il ya des activités a exécuter. On d'autre terme à chaque unité de temps on vérifie si toutes les conditions requise pour qu'une activité soit exécutés sont satisfaites. Si elles le sont, on exécute l'action correspondante sinon on augmente l'horloge d'une unité et on recommence, ceci est une simulation à deux phases. On ne définit que les activités C pas d'activités B, donc toutes les activités on un test.

2. Approche orientée évènements

Lorsque on utilise cet approche l'horloge est avancé au prochain évènement. On doit garder une liste chronologique contenant le moment où il doit se produire et les changements d'état correspondant lorsqu'on change l'état, on engendre un nouveau évènement qui doit à son tour être placé dans la liste.

Lorsqu'un évènement est passé on l'on lève de la liste, c'est une simulation à deux phases.

Il n'ya pas des activités **C** seulement les activités **B** appelées évènements.

3. Approche par interaction des processus

Cette approche se caractérise par:

- La présence des séquences d'évènements ou des activités pour un type d'objet, définie sous forme de processus.
- La description du fonctionnement du système complet par macro-représentation.
- La gestion des conflits et la synchronisation entre processus par des règles d'interruption et de reprise.

1.5.2. Exemple de simulation

Un exemple classique de système à événements discrets est une file d'attente, c.-à-d. un modèle qui représente l'accès séquentiel d'un ensemble d'utilisateurs (par exemple clients) à un nombre limité de ressources (par exemple les guichets d'une banque).

Le terme générique client peut être utilisé pour dénoter des personnes, des machines, des voitures, des patients, des paquets des données, des emails.

Le terme générique service peut être utilisé pour dénoter des employés, des dépanneurs, des mécaniciens, des docteurs, des routes, des grues.

La théorie des files d'attente et/ou la simulation peuvent être utilisées pour prédire la performance du système (par exemple la moyenne du temps d'attente d'un client) en fonction des paramètres (par exemple le nombre des services, la loi des arrivées, la loi des services) qui sont sous le contrôle du concepteur du système.

1.6. Simulation de Monte Carlo

Le terme méthode de Monte-Carlo, ou méthode Monte-Carlo (MC), désigne toute méthode visant à calculer une valeur numérique en utilisant des procédés aléatoires, c'est-à-dire des techniques probabilistes. Le nom de ces méthodes, qui fait allusion aux jeux de hasard pratiqués à Monte-Carlo, a été inventé en 1947 par Nicholas Metropolis, et publié pour la première fois en 1949 dans un article coécrit avec Stanislaw Ulam.

Les méthodes de Monte-Carlo sont particulièrement utilisées pour calculer des intégrales en dimensions plus grandes que 1 (en particulier, pour calculer des surfaces et des volumes). Elles sont également couramment utilisées en physique des particules, où des simulations probabilistes permettent d'estimer la forme d'un signal ou la sensibilité d'un

détecteur. La comparaison des données mesurées à ces simulations peut permettre de mettre en évidence des caractéristiques inattendues, par exemple de nouvelles particules.

La méthode de simulation de Monte-Carlo permet aussi d'introduire une approche statistique du risque dans une décision financière. Elle consiste à isoler un certain nombre de variables-clés du projet, tels que le chiffre d'affaires ou la marge, et à leur affecter une distribution de probabilités.

Pour chacun de ces facteurs, un grand nombre de tirages aléatoires est effectué dans les distributions de probabilité déterminées précédemment, afin de trouver la probabilité d'occurrence de chacun des résultats.

1.6.1. La méthode de Monte Carlo

1.6.1.1. Présentation de la méthode

La méthode de Monte Carlo [11] est très utile pour approcher numériquement des calculs non explicites. Il peut s'agir par exemple de résoudre des équations aux dérivées partielles, des systèmes linéaires ou des problèmes d'optimisation.

Cette méthode consiste à approcher $I = \int_{\alpha}^{\beta} f(x) dx$ d'une fonction définie sur $]\alpha, \beta[$, continue et éventuellement non borné sur cette intégrale I .

En écrivant I sous la forme $I = (\beta - \alpha) E(f(U))$ et on approche $(f(u))$.

X est une variable aléatoire qui suit la loi uniforme sur $]\alpha, \beta[$ et $E(f(U))$ représente l'espérance mathématique de la variable aléatoire $f(u)$.

$$\text{On a } E(f(U)) = \int_{\alpha}^{\beta} f(u) * f_U(x) dx = \frac{1}{\beta - \alpha} \int_{\alpha}^{\beta} f(x) dx = \frac{I}{\beta - \alpha}.$$

Si l'intégrale I est généralisée veut dire que f est non bornée alors, on suppose qu'elle est convergente; et on supposera aussi que $\int_{\alpha}^{\beta} f^2(x) dx$ est convergente.

Pour ce faire, on utilise la loi des grands nombres: si $(U_i)_{i \in \mathbb{N}^*}$ est une suite de variable aléatoire mutuellement indépendantes et indépendantes de U , de loi U $[\alpha, \beta]$ alors $\lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{i=1}^n f(u_i) = E(f(U))$.

Soit $I_n = \frac{1}{n} \sum_{i=1}^n f(u_i)$ est une approximation de $\frac{I}{\beta - \alpha}$ si $(\beta - \alpha) * I_n$ est l'approximation de l'intégral I . Ou u_1, u_2, \dots, u_n sont des nombres tirées aléatoirement de

$U [\alpha, \beta]$.

1.6.1.2. Vitesse de convergence

La vitesse de convergence [15] d'une estimation par la méthode de Monte Carlo est donnée par le théorème central limite.

D'après le Théorème central limite $\frac{\frac{1}{n} \sum_{i=1}^n f(U_i) - E(f(U))}{\frac{\sigma}{\sqrt{n}}} \sim N(0,1)$.

Où G est une variable aléatoire gaussienne centrée de variance 1. Par conséquent pour n suffisamment grand, l'écart $\varepsilon_n = \frac{1}{n} \sum_{i=1}^n f(U_i) - E(f(U))$ satisfait

$$P\left(\frac{\sqrt{n}}{\sigma} \varepsilon_n \leq 1,96\right) = 0.95$$

Et donc $|\varepsilon_n|$ est avec une quasi-certitude de 95% inférieur à $\frac{1.96}{\sqrt{n}}$. Par suite un intervalle de confiance pour $E(f(U))$ au niveau 0.95 est $\left[\frac{1}{n} \sum_{i=1}^n f(U_i) - \sigma \frac{1.96}{\sqrt{n}}, \frac{1}{n} \sum_{i=1}^n f(U_i) + \sigma \frac{1.96}{\sqrt{n}}\right]$.

En général σ ne sera pas calculable et on l'approximera par une méthode de MC puisque

$$\frac{1}{n} \sum_{i=1}^n E(f^2(U_i)) - \left(\frac{1}{n} \sum_{i=1}^n (E(f(U_i)))^2\right) \sim \sigma^2 \text{ presque sûrement}$$

Ce calcul sera mené en même temps que celui de l'espérance.

La vitesse de convergence est donc de l'ordre de $\frac{\sigma}{\sqrt{n}}$. Cette vitesse peut paraître faible en petite dimension [10].

1.6.1.3. Calcul de la variance

En note que $\sigma^2 = Var(f(U)) = E[f(U) - E(f(U))]^2 = E(f^2(U)) - (E(f(U)))^2$.

On générale la valeur de σ^2 n'est pas connue, l'approche de sa valeur en utilisant le rapprochement des $E(f(U))$ en fonction de la convergence suivant:

$$0 \leq \frac{1}{n} \sum_{i=1}^n (f(u_i))^2 - I_n^2 = \frac{1}{n} \sum_{i=1}^n (f(U_i) - I_n)^2 \rightarrow \sigma^2 \text{ quand } n \rightarrow +\infty$$

1.7.1.4. L'approximation des intégrales

Pour n assez grand, la méthode le MC fournit [7]:

- $\delta = (\beta - \alpha)I_n$ est une approximation de l'intégrale I .
- $\hat{\sigma} = \sqrt{\frac{1}{n} \sum_{i=1}^n f^2(U_i) - I_n^2}$ une approximation de l'écart type σ .
- $E = (\beta - \alpha) \hat{\sigma} \frac{1.96}{\sqrt{n}}$ est une approximation de l'erreur $|\delta - I|$ à 95%.

1.6.1.5. Exemple

Soit $f(x) = e^x$ est définie sur un intervalle ouvert et borné $]0,1[$ et f est continue, on peut donc utiliser l'approximation de MC .

Alors $\int_0^1 e^x dx$ existe et $\int_0^1 (e^x)^2 dx$ existe.

On a $I_n = \frac{1}{n} \sum_{i=1}^n f(u_i)$ et en utilisant la suite de nombre aléatoire : 0.05 - 0.89 - 0.66 - 0.34.

on obtient $I_4 = \frac{1}{4} (e^{0.05} + e^{0.89} + e^{0.66} + e^{0.34}) = 1.706$.

- $\delta = (1 - 0)I_4 = I_4 = 1.706$ est une approximation de I et de $E(f(u))$.
- $\hat{\sigma} = \frac{1}{4} [(e^{0.05})^2 + (e^{0.89})^2 + (e^{0.66})^2 + (e^{0.34})^2] - (1.706)^2 = 3.19 - (1.706)^2 = 0.28$ est une approximation de σ .
- $E = (1 - 0) \frac{\sqrt{0.28} * 1.96}{2} = 0.52$ est une approximation de l'erreur à 95%.

1.6.1.6. Avantage de la méthode MC

- construire un échantillon de loi connue permettant d'étudier les paramètres d'un échantillon de loi inconnue.
- donner une estimation d'un paramètre, d'une constante, d'une intégrale.

1.7. L'échantillonnage descriptif

L'échantillonnage descriptif (ED) a été proposé par E. Saliby (1980 et 1990) est basé sur une sélection totalement déterministe de l'ensemble de l'échantillon d'entrée de valeurs et de leur permutation aléatoire. Soit discret ou continu ou même une distribution mixte peut être représentée, à condition que l'inverse respectif de la fonction de distribution n'est pas disponible.

Cette fonction inverse est toujours définie, bien que, dans la plupart des cas, une approximation numérique peut être nécessaire, comme dans le cas d'une distribution normale

(Ramberg et Schmeiser 1972). Formellement, l'échantillonnage descriptif lorsque la taille N de l'échantillon est connue, les valeurs d'échantillonnage sont définies pour l'entrée variable aléatoire X en utilisant la méthode de transformée inverse, tel que :

$$X_i = F^{-1}(r_i) \quad \text{pour } i = 1, \dots, N$$

F^{-1} est la distribution d'entrée cumulée inverse et l'ensemble des nombres réguliers $\{r_i, i = 1, 2, \dots, N\}$ est uniformément distribué sur $[0, 1[$ est définie par ce qui suit [8]:

$$r_i = \frac{i-0.5}{N} \quad i = 1, 2, \dots, N$$

les variables d'entrées sont bien représenté avec ED par rapport a la méthode de MC.

Exemple 1:

Estimer la moyenne de la loi exponentielle de paramètre $\lambda = 2$, $r_i = \frac{i-0.5}{N}$ avec $N = 6$.

i	r_i	X_i
1	0.083	1.244
2	0.25	0.693
3	0.417	0.437
4	0.583	0.270
5	0.75	0.144
6	0.917	0.043
Total		2.831

Table1.1. Résultats obtenues de la méthode ED

l'estimateur de la moyenne est: $\hat{\mu} = \frac{1}{N} \sum_{i=1}^N X_i = \frac{2.831}{6} = 0.742$.

Exemple 2:

Estimer la moyenne de la loi uniforme sur $[0,1]$, $r_i = \frac{i-0.5}{N}$ avec $N = 4$

i	r_i	X_i
1	0.125	0.125
2	0.375	0.375
3	0.675	0.675
4	0.875	0.875
Total		2.05

Table 1.2. Résultats obtenues par la méthode ED

L'estimateur de la moyenne est: $\hat{\mu} = \frac{1}{N} \sum_{i=1}^N X_i = \frac{2.05}{4} = 0.5125$.

Les avantages :

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

Les inconvénients :

- ❖ La taille de l'échantillon doit être connue au début de la simulation.

1.8. Conclusion

Des alternatives à la méthode de Monte Carlo ont été développées, non seulement pour accélérer la convergence des estimations, mais aussi pour assurer que les domaines de variation des paramètres d'entrée soient bien échantillonnés.

En effet, placer quelque point au hasard dans un espace de grande dimension conduit le plus souvent à une mauvaise couverture de certaines dimensions [16]. Quelques méthodes développées pour contrecarrer ce problème ont été celle des hypercubes latin et échantillonnage descriptif amélioré qui seront présentés dans les chapitres suivants.

Chapitre **II**

Echantillonnage descriptif amélioré

Et par hypercube latin

2.1. Introduction

Dans ce chapitre, nous allons détailler le principe de deux méthodes d'échantillonnage .

La première est basée sur l'échantillonnage descriptif amélioré et la deuxième sur l'échantillonnage par l'hypercube latin.

La procédure d'échantillonnage descriptif amélioré (EDA) est introduit par [Tari (2002 et 2005)] consiste à réduire le risque d'un biais d'échantillonnage qui peut être généré par ED [13].

L'échantillonnage par Hypercube latine (LHS) est proposé en 1979, McKay et al. comme une technique de réduction de la variance dans lequel la sélection de valeurs d'échantillon est fortement contrôlé, bien que toujours en leur permettant de varier. La base de LHS est une stratification complète de la distribution échantillonné avec une sélection aléatoire à l'intérieur de chaque strate. C'est un type d'échantillonnage aléatoire stratifié où les valeurs de l'échantillon sont mélangées aléatoirement parmi les différentes variables [1].

2.2. Echantillonnage descriptif amélioré

2.2.1. Principe d'EDA

Cette méthode est principalement concerné par un bloc de nombres premiers qui doivent être situés à l'intérieur d'un générateur visant à distribuer des échantillons de la variable d'entrée de taille premier p_j, p_{j+1}, \dots où j est choisit aléatoirement.

Les valeurs sont générées à la demande de la simulation. Nous arrêtons le processus de génération lorsque la simulation se termine. [8].

$$\underbrace{\underbrace{p_1 = 79}_{1^{ere} \text{ sous-échantillon}} \quad \underbrace{p_2 = 11}_{2^{ème} \text{ sous-échantillon}} \quad \dots \dots \dots \quad \underbrace{p_m = 7}_{m^{ème} \text{ sous-échantillon}}}_{EDA \text{ échantillon de taille } n}$$

Figure 2.1 : Illustration de l'échantillon EDA de taille $n = \sum_j^m p_j$ [2].

$$\underbrace{\underbrace{p_1 = 79}_{1^{ere} \text{ sous-histoire}} \quad \underbrace{p_2 = 11}_{2^{ème} \text{ sous-histoire}} \quad \dots \dots \dots \quad \underbrace{p_m = 7}_{m^{ème} \text{ sous-histoire}}}_{un \text{ bloc} = EDA \text{ histoire}}$$

Figure 2.2 : Illustration d'une histoire EDA [2].

Un échantillon descriptif amélioré de taille $n = \sum_j^m p_j$ est défini par la m sous-échantillons successifs appelés également des échantillons de taille descriptives $p_j, j = 1, \dots, m$ comme illustré sur la Figure 1.

Une expérience de simulation d'une histoire EDA est définie par m sous-histoire où la $j^{\text{ème}}$ sous-histoire utilise un échantillon descriptif de la taille p_j comme illustré sur la figure. 2.

Une expérience de simulation de N réplication définie par l'utilisation de N blocs de m_1, m_2, \dots, m_N sous-échantillons de nombres premiers tailles $p_{jq}, j = 1, 2, \dots, m_q$ et $q = 1, 2, \dots, N$ [2].

2.2.2. Générer des valeurs avec EDA

Le procédé général de la transformée inverse produit des valeurs de la variable aléatoire d'entrée X qui sont donnés par :

$$X_i = F^{-1}(r_i) \text{ pour } i = 1, 2, \dots, p_j, j = 1, 2, \dots, m$$

Où F^{-1} est la distribution cumulative d'entrée inverse et r_i suit $U [0,1]$ tel que [8]

$$r_i = \frac{i-0.5}{p_j} \quad i = 1, 2, \dots, p_j, \quad j = 1, 2, \dots, m$$

2.2.3. Algorithme d'EDA

Structure de données

On définit chaque variable aléatoire d'entrée comme suit:

p est un nombre premier pour définir la taille de la sous ensemble des numéros réguliers.

R est un tableau des nombre réels contenant le sous ensemble de numéros régulière.

ip nombre entier désignant le premier élément r disponible à tirer. Si $ip = 1$, aucun élément n'a encore été établi. Si $ip > p$, un sous-ensemble complet des numéros réguliers a déjà été établi.

début

(a) L'initialisation de l'expérience.

(a₁) Avant de lancer une simulation, générer une séquence de nombres premiers distincts.

(a₂) Choisissez au hasard sans remplacement un nombre premier p de cette séquence.

(a₃) Générer le sous-ensemble de nombres régulière r_i , $i = 1, 2, \dots, p$ et les stocker dans un tableau R.

(b) Initialisation pour la sous-histoire. Au début de chaque sous-histoire, laisser $ip: = 1$.

(c) Échantillonnage sans remplacement durant la sous histoire:

(c₁) Si $ip > p$ aller à (d).

(c₂) Générer aléatoirement un entier $iaux \in [ip, p]$.

(c₃) Echanger $r(ip)$ avec $r(iaux)$.

(c₄) Générer une observation X_i .

Si une valeur de la sous-ensemble descriptive n'est pas nécessaire arrêter et recueillir les résultats nombre premier trouve avant utilisé aller à (e).

(c₅) Sinon, laisser $ip: = ip + 1$ et aller à (c₁).

(d) Recueillir les résultats après chaque sous-histoire et aller à (a₂).

(e) Recueillir les résultats après chaque exécution [8].

Fin.

2.2.4. La différence entre EA, ED et EDA

EA, ED et EDA sont toutes des méthodes d'échantillonnage qui peuvent être utilisés dans une étude de simulation pour générer des variables d'entrée connues. Chaque méthode a ses propres caractéristiques, mais dans toutes les méthodes d'échantillonnages, les valeurs d'entrée d'un échantillon de taille n est généré par la formule suivante:

$$X_i = F^{-1}(u_i) \text{ pour } i = 1, 2, \dots, n$$

Où u_i sont des:

- Nombres aléatoires générées avec un certain générateur de nombres aléatoires pour l'utilisation d'EA.

- Numéros réguliers générés par

$$r_i = \frac{i - 0.5}{n}$$

Lors de l'utilisation ED

- Numéros réguliers générés comme dans ED, mais à partir d'échantillons successifs de taille nombre premier p_j lorsque vous utilisez EDA. La taille de l'échantillon [8] EDA n est

$$\sum_j^m p_j.$$

2.2.6. Exemple

Le Sous-ensembles de nombres premiers de taille $p_1=11$, $p_2 = 5$ et $p_3= 7$ pour une distribution uniforme de paramètre 0 et 1.

i	r_i	x_i	I	r_i	x_i	i	r_i	x_i
1	0.045	0.045	1	0.100	0.100	1	0.071	0.071
2	0.136	0.136	2	0.300	0.300	2	0.214	0.214
3	0.227	0.227	3	0.500	0.500	3	0.357	0.357
4	0.318	0.318	4	0.700	0.700	4	0.500	0.500
5	0.409	0.409	5	0.409	0.409	5	0.643	0.643
6	0.500	0.500				6	0.786	0.786
7	0.591	0.591				7	0.929	0.929
8	0.682	0.682						
9	0.773	0.773						
10	0.864	0.864						
11	0.955	0.955						

Table 2.1. Illustration de la méthode EDA.

	p₁	p₂	p₃	Moyenne global
moyenne	0.516	0.402	0.5	0.473
biais	0.016	-0.098	0	-0.027

Table 2.2. La moyenne observée d'une distribution uniforme en utilisant trois sous-ensemble de nombre premier.

Une fois que le sous-ensemble de numéros réguliers r_i est calculé, elle est ensuite utilisée comme requis par la simulation. Lorsque la simulation est terminée, nous nous arrêtons de produire des valeurs de variable d'entrée et par conséquent nous cessons la production de nombres premiers.

EDA méthode est illustré dans le tableau 1 prise $p_1 = 11$, $p_2 = 5$ et $p_3 = 7$ pour la distribution uniforme de paramètre 0 et 1 avec une moyenne $\frac{1}{2}$ obtenue par $x_i = r_i$. Par conséquent, dans un

bloc donné, l'estimation de la moyenne et le calcul de biais sont obtenues comme suit:

$$Y = \frac{1}{3} \sum_{i=1}^3 y_i = 0.473$$

$$\text{Biais}(y) = \frac{1}{3} \sum_{i=1}^3 \text{Biais}(y_i) = -0.027$$

$$n = \sum_{i=1}^3 p_i = 23$$

2.3. Echantillonnage par Hypercube Latin

2.3.1. Méthode Hypercube Latin

La méthode hypercube latin (LHS) (McKay et al., Stein) consiste à répartir les points de l'échantillon beaucoup plus uniformément que par Monte Carlo, sur toute l'étendue du domaine de chaque variable d'entrée. Pour cela, l'intervalle de chaque entrée X_i est découpé en N segments de probabilité égale à $1/N$; puis une valeur est tirée aléatoirement dans chacun de ces segments. Ainsi, une fois qu'un point a été tiré aléatoirement dans un segment de X_i , aucun nouveau point ne peut y être placé. Les N valeurs ainsi obtenues pour X_1 sont combinées de manière aléatoire aux N valeurs obtenues pour X_2 et forment ainsi une matrice de taille $N \times 2$.

De même, les N valeurs obtenues pour X_3 sont accolées à cette matrice et ainsi de suite jusqu'à la k -ième variable.

On distingue l'échantillonnage par hypercube latin de deux notions très voisines définies ci-dessous : le plan en hypercube latin et un échantillon en hypercube latin centrée uniforme [15].

2.3.2. Quelques définitions sur LHD

Définition 1:

Un plan en hypercube latin (LHD, pour Latin Hypercube Design) pour N simulation et K dont chaque colonne (sauf éventuellement la première) est une permutation aléatoire de $1, \dots, N$. On le note LHD(N, K) [15].

Définition 2 :

Un échantillon en hypercube latin centré uniforme (noté cLHS, pour centered Latin

Hypercube Sample) pour N simulation et K facteurs est une matrice $N \times K$ dont chaque colonne (sauf éventuellement la première) est une permutation aléatoire de $(1-0.5)/N, \dots, (N-0.5)/N$.

On le note cLHS (N,K) [15].

2.3.3. Principe de la méthode

Pour appliquer la méthode de l'hypercube latin (LHS), on peut procéder de la manière suivant:

Etape 1 : Générer un plan en hypercube latin $D=LHD(N,K)$ par K permutations aléatoires de $1, \dots, N$;

Etape 2 : Soit obtenir un hypercube latin centré cLHS(N,K) en calculant $D'=(D-0.5)/N$;

Etape 2bis soit obtenir un hypercube latin LHS(N,K) en soustrayant à chaque élément de D le résultat d'un tirage aléatoire indépendant dans la loi uniforme sur $[0,1/N]$. Dans le cas général de variable aléatoires X_1, \dots, X_K mutuellement indépendant et de fonction de répartition continues inversibles F_1, \dots, F_K le $i^{ème}$ échantillon de la $j^{ème}$ variable peut être créé par

$$x_j^{(i)} = F_j^{-1} \left(\frac{\pi_j^{(i)} - \varepsilon_j^{(i)}}{N} \right)$$

Où $(\pi_j)_{j=1, \dots, k}$ sont des permutations aléatoires indépendantes des entiers $\{1, 2, \dots, N\}$, et $\xi_j^{(i)}_{i=1, \dots, N}$ sont des réalisations $U[0,1]$ indépendantes entre elles, et indépendantes de π_j .

Nous appellerons centrage l'étape 2 et brouillage l'étape 2bis (scrambling en anglais). Les résultats sont illustrés dans la figure 2.1, obtenue par centrage et brouillage du LHD du tableau 2.3.

Indice i	1	2	3	4	5	6	7	8	9	10
X_1	1	2	3	4	5	6	7	8	9	10
X_2	9	3	1	10	6	5	7	2	4	8

Table 2.3. Plan en hypercube latin LHD(10,2) (transposé).

La méthode LHS garantit une couverture uniforme du domaine de chaque entrée évitant de sous-échantillonner certains segments et l'inverse d'un sur-échantillonner d'autres. Ainsi, elle permet de faire converger plus rapidement des statistiques associées au comportement global de la sortie Y que la méthode de Monte Carlo simple. Par contre, la méthode LHS au strict ne contrôle absolument pas la qualité de la répartition conjointe des échantillons lorsque l'on considère deux ou plus variable d'entrée [15].

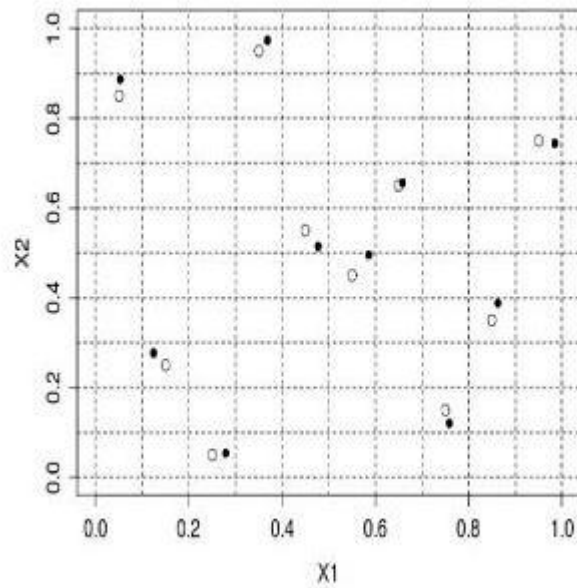


Figure 2.3. Hypercube latin de taille 10 pour deux variables d'entrée X_1 et X_2 . Points blancs : hypercube latin centrée (cLHS). points noirs : hypercube latin non centrée (LHS).

Exemple 2 :

Nous allons illustrer la construction d'un échantillonnage LHS dans le cas de la dimension 2 avec 6 points d'échantillonnage. Dans cet exemple, nous avons donc 2 variables X_1 et X_2 .

Le domaine de chacune de ces 2 variables est $[-3 ; 3]$. Comme le nombre de points désire dans l'échantillon est 6, on divise ces 2 intervalles en 6 sous-intervalles. Ceux-ci sont $[-3 ; -2]$, $[-2 ; -1]$, ..., $[2 ; 3]$ dans chaque direction. La phase suivante pour obtenir l'échantillonnage de l'Hypercube Latin est de sélectionner des valeurs spécifiques de X_1 et X_2 dans les 6 intervalles de chacune des dimensions. Ainsi, X_1 et X_2 possèdent maintenant 6 valeurs spécifiques.

Ensuite, les valeurs sélectionnées de X_1 et X_2 sont aléatoirement couplées afin d'obtenir les 6 coordonnées des points de l'échantillonnage souhaités.

Dans le premier concept de l'Hypercube Latin, les couples étaient conçus en utilisant une permutation aléatoire des n premiers entiers.

Dans notre exemple précédent, on considère deux permutations aléatoires des entiers (1, 2, 3, 4,5,6) comme suit :

Permutation numéro 1 : (3, 1, 5, 2, 4, 6)

Permutation numéro 2 : (2, 4, 1, 3, 5, 6)

En considérant que les nombres de ces permutations correspondent à des numéros d'intervalles pour X1 (permutation 1) et X2 (permutation 2), les couples suivants d'intervalles peuvent être formés :

Point	Numéro d'intervalle pour X1	Numéro d'intervalle pour X2
1	3	2
2	1	4
3	5	1
4	2	3
5	4	5
6	6	6

Pour le premier point, les coordonnées sont déterminées en sélectionnant la valeur spécifique de X1 de l'intervalle 3 ([-1 ; 0]) et en l'associant à la valeur spécifique de X2 de l'intervalle 2 ([-2 ; -1]). Les coordonnées des autres points sont obtenues de la même manière. La figure suivante (figure 2) montre un exemple d'échantillonnage d'Hypercube Latin.

On constate effectivement sur cet exemple que, grâce à cette technique, on obtient un seul point dans chaque sous-intervalle de X1 et de X2 [4].

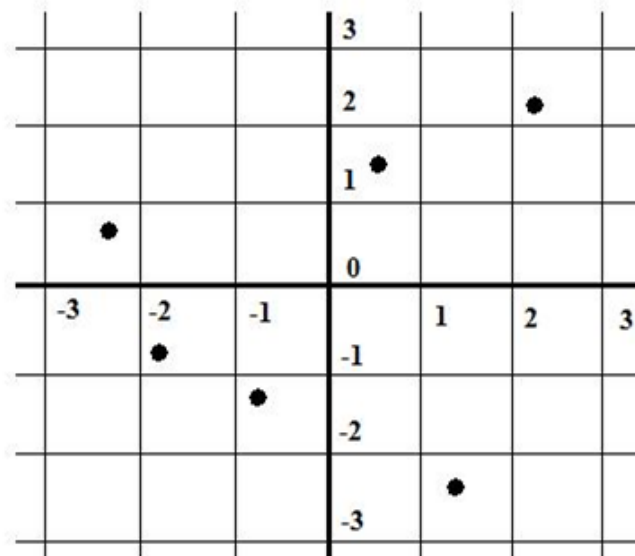


Figure 2.4 . Exemple d'échantillonnage LHS pour 6 points en dimension 2

2.3.4. Avantages

- ❖ donne des échantillons uniformément répartis dès un faible nombre de simulation.

2.3.5. Inconvénients

- ❖ le choix de N est parfois délicat.
- ❖ temps de calcul sont beaucoup plus importants que lors de simulations pseudo aléatoires classiques.
- ❖ programmation délicate : parcourir tous les sous ensemble de $[0,1]$ de manière efficace.
- ❖ découpage de intervalles => simuler au moins une valeur par intervalle.

2.4. Conclusion

Nous avons vu dans ce chapitre toutes les définitions concernant les deux méthodes LHS et EDA, mais on veut savoir la quelle parmi eux est meilleur pour cela nous proposons une application dans le chapitre suivant qui va nous aider à effectuer un test pour décider laquelle des deux méthodes est meilleure.

Chapitre III

Les systèmes de files d'attente (M/M/1,
M/G/1)

3.1. Introduction

Le modèle général d'un phénomène d'attente, appelé aussi système d'attente, peut être résumé comme suit : des «clients» arrivent à un certain endroit et réclament un certain service. Les instants d'arrivée et les durées de service sont généralement des quantités aléatoires. Si un poste de service est libre, il prend dans une file d'attente dans laquelle les clients se rangent suivant leur ordre d'arrivée.

Un système d'attente comprend donc un espace de service avec une ou plusieurs stations de service montées en parallèle, et un espace d'attente dans lequel se forme une éventuelle file. Des phénomènes d'attente se manifestent sous des formes multiples dont voici quelques exemples :

- Arrivée de voitures vers une station-service,
- Fonctionnement d'un central téléphonique,
- Vente de billets auprès d'un guichet,
- Réparation de machines défectueuses par un mécanicien,
- Exécution des tâches dans un centre de calcul,
- Atterrissages des avions sur un aéroport,

La théorie des systèmes d'attente a comme objectif d'en étudier les structures et de calculer des valeurs caractéristiques permettant de décrire les performances d'un système [3].

3.2. Classification des systèmes d'attente

Pour identifier un système d'attente, on a besoin des spécifications suivantes.

- La nature stochastique du processus des arrivées (ou flux d'entrée), qui est défini par la distribution des intervalles séparant deux arrivées consécutives.
- La distribution du temps aléatoire de service.
- Le nombre s de stations de service qui sont montées en parallèle ou en série. On admet généralement que les temps de service correspondants suivent la même distribution et que les clients qui arrivent forment une seule file d'attente.
- La capacité N du système. Si $N < \infty$, la file d'attente ne peut dépasser une longueur de $N - s$ unités. Dans ce cas, certains clients qui arrivent vers le système n'ont pas la

possibilité d'y entrer.

On suppose par ailleurs que toutes les variables aléatoires introduites pour décrire un phénomène d'attente sont indépendantes.

Pour la classification des systèmes d'attente, on recourt à une notation symbolique comprenant quatre symboles rangés dans l'ordre

$$A / B / s / N$$

Où

A = distribution des temps entre deux arrivées successives,

B = distribution des durées de service,

s = nombre de postes de service en parallèle

N = capacité du système.

Le dernier symbole sera toutefois supprimé si $N = \infty$. Pour spécifier les distributions A et B, on introduit les symboles suivants :

M = distribution exponentielle (qui vérifie donc la propriété de Markov),

E_k = distribution d'Erlang d'ordre k,

G = distribution générale,

En plus des notations introduites ci-dessus, on utilisera les grandeurs suivantes :

$1/\lambda$ = intervalle moyen séparant deux arrivées consécutives, d'où

λ = taux des arrivées,

$1/\mu$ = durée moyenne de service, d'où

μ = taux de service.

Dans le cas particulier des distributions exponentielles, les taux λ et μ sont identiques aux paramètres de ces distributions [3].

3.3. Intervalle entre deux événements

Soit $\{N(t) ; t \geq 0\}$ un processus de Poisson de paramètre λ et T_n la durée séparant le (n-1)-ième et le n-ième événement.

Théorème : Les temps d'attente T_n d'un processus de Poisson de paramètre λ sont des variables aléatoires indépendamment et distribuées identiquement selon une loi exponentielle de paramètre λ [3].

3.4. Caractéristiques d'un système d'attente

3.4.1. Définitions, formules de Little

A partir de la distribution stationnaire de processus $\{X(t) ; t \geq 0\}$, on peut calculer d'autres valeurs caractéristiques d'un système d'attente telles que :

- $L = E(X)$ = nombre moyen de clients dans le système,
- L_q = nombre moyen de clients dans la file d'attente,
- W = temps de séjour moyen d'un client dans le système,
- W_q = temps d'attente moyen d'un client,
- W_q^* = temps d'attente moyen d'un client qui est obligé d'attendre.

Ces valeurs permettent de juger du comportement opérationnel d'un système d'attente. Elles ne sont pas indépendantes les unes des autres, mais sont liées par les relations suivantes :

- $L = \lambda_e W$
- $L_q = \lambda_e W_q$
- $W = W_q + 1/\mu$
- $L = L_q + \lambda_e / \mu$

Où λ_e est le taux d'entrée des clients dans le système. Si la capacité du système est illimitée, on a $\lambda_e = \lambda$. Dans le cas contraire, certains des clients doivent s'en aller sans être servis d'où $\lambda_e < \lambda$.

Ces quatre relations sont valables dans des conditions assez générales, c'est-à-dire pour des systèmes d'attente du type $G / G / s / N$. Les deux premières relations, qui sont connues sous le nom de formules de Little [3], peuvent être vérifiées à l'aide du raisonnement intuitif suivant. D'après nos définitions, W représente le temps moyen qu'un client passe dans le système; d'autre part, λ_e est le nombre moyen de clients entrés dans le système par unité de temps. Il en résulte qu'à l'instant où un client quitte le système, il y a derrière lui en moyenne

$\lambda_e W$ clients dans le système, d'où $L = \lambda_e W$.

3.5. Système de file d'attente M/M/1

C'est le plus simple des systèmes de files d'attente. Dans ce système, le flux des arrivées est poissonien, de paramètre λ , et la durée de service est exponentielle, de paramètre μ . La capacité d'attente est illimitée et il y a une seule station de service [3].

3.5.1. Régime stationnaire du système

Grâce aux propriétés fondamentales du processus de poisson et de la loi exponentielle, nous avons pour un petit intervalle de temps Δt les probabilités suivantes:

- $P(\text{exactement 1 arrivée pendant } \Delta t) = \lambda \Delta t + 0(\Delta t)$
- $P(\text{aucune arrivée pendant } \Delta t) = 1 - \lambda \Delta t + 0(\Delta t)$
- $P(\text{2 arrivées ou plus pendant } \Delta t) = 0(\Delta t)$
- $P(\text{exactement 1 départ pendant } \Delta t / X(t) \geq 1) = \mu \Delta t + 0(\Delta t)$
- $P(\text{aucun départ pendant } \Delta t / X(t) \geq 1) = 1 - \mu \Delta t + 0(\Delta t)$
- $P(\text{2 départs ou plus pendant } \Delta t) = 0(\Delta t)$

Ces probabilités ne dépendent ni du temps t ni de l'état $X(t)$ dans lequel le système se trouve.

Nous désignons maintenant par $p_{ij}(\Delta t)$ la probabilité (conditionnelle) que le processus $X(t)$ fasse une transition de i vers j pendant la durée Δt :

$$p_{ij}(\Delta t) = P(X(t + \Delta t) = j \mid X(t) = i)$$

où $i, j = 0, 1, 2, \dots$. Ces probabilités de transition ne dépendent pas de l'instant t .

Puisque les arrivées et les départs se produisent indépendamment les uns des autres, on a

- $p_{n,n+1}(\Delta t) = \lambda \Delta t (1 - \mu \Delta t) + 0(\Delta t) = \lambda \Delta t + 0(\Delta t) \quad (n \geq 0)$
- $p_{n,n}(\Delta t) = \lambda \Delta t * \mu \Delta t + (1 - \lambda \Delta t)(1 - \mu \Delta t) + 0(\Delta t)$
 $= 1 - (\lambda + \mu) \Delta t + 0(\Delta t) \quad (n \geq 1)$

- $p_{0,0}(\Delta t) = 1 - \lambda\Delta t + o(\Delta t)$
- $p_{n+1,n}(\Delta t) = (1 - \lambda\Delta t)\mu\Delta t + o(\Delta t) = \mu\Delta t + o(\Delta t) \quad (n \geq 0),$

tandis que

$$p_{n,m}(\Delta t) = o(\Delta t) \text{ pour } |m - n| \geq 2.$$

A partir d'un état donné n , le processus ne peut donc que passer dans l'un des états voisins $n - 1$ et $n + 1$.

Pour calculer les probabilités d'états $p_n(t) = P(X(t) = n)$, nous pouvons écrire, d'après le théorème des probabilités totales et pour $n \geq 1$:

$$p_n(t + \Delta t) = \sum_{i=0}^{\infty} p_i(t)p_{in}(\Delta t)$$

$$p_{n-1}(t)\lambda\Delta t + p_n(t)[1 - (\lambda + \mu)\Delta t] + p_{n+1}(t)\mu\Delta t + o(\Delta t)$$

D'où

$$\frac{p_n(t + \Delta t) - p_n(t)}{\Delta t} = -(\lambda + \mu)p_n(t) + \lambda p_{n-1}(t) + \mu p_{n+1}(t) + o(\Delta t)\Delta t.$$

En faisant tendre Δt vers 0, on trouve

$$p'_n(t) = -(\lambda + \mu)p_n(t) + \lambda p_{n-1}(t) + \mu p_{n+1}(t).$$

Une adaptation évidente de ce raisonnement nous donne $p'_0(t) = -\lambda p_0(t) + \mu p_1(t)$.

Ces équations sont connues sous le nom d'équations différentielles de Kolmogorov; elles permettent en principe de calculer les probabilités d'états $p_n(t)$ si l'on connaît en plus les conditions initiales du processus, c'est-à-dire la distribution de $X(0)$. On peut ainsi déterminer le régime transitoire du processus stochastique $\{X(t); t \geq 0\}$ à partir des taux de transition λ et μ [3].

3.5.2. Régime stationnaire

Lorsque $t \rightarrow \infty$ dans le système d'équations de Kolmogorov, on peut montrer que les limites

$$\lim_{t \rightarrow \infty} p_n(t) = p_n$$

existent et sont indépendantes de l'état initial du processus et que $\lim_{t \rightarrow \infty} p'_n(t) = 0$ ($n = 0, 1, 2, \dots$).

A la place d'un système d'équations différentielles, on obtient alors un système d'équations linéaire et homogènes

$$\mu p_1 = \lambda p_0$$

$$\lambda p_{n-1} + \mu p_{n+1} = (\lambda + \mu)p_n \quad (n = 0, 1, 2, \dots).$$

auxquelles il faut ajouter la condition

$$\sum_{n=0}^{\infty} p_n = 1.$$

Sa résolution est élémentaire; en additionnant les $n + 1$ premières équations, on obtient

$$\mu p_{n+1} = \lambda p_n$$

d'où

$$p_n = p_0 \left(\frac{\lambda}{\mu}\right)^n \quad (n = 0, 1, 2, \dots)$$

En se servant du fait que les p_n définissent une distribution de probabilité, on obtient finalement

$$p_n = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^n$$

à condition que $\lambda < \mu$; le régime stationnaire du système d'attente M/M/1 est donc gouverné par la loi géométrique de paramètre $\left(1 - \frac{\lambda}{\mu}\right)$.

3.5.3. Caractéristiques du système M / M / 1

a) Le nombre moyen de clients dans le système

$$\begin{aligned} L = E(X) &= \sum_{n=0}^{\infty} n p_n = \sum_{n=0}^{\infty} n (1 - \rho) \rho^n = (1 - \rho) \sum_{n=0}^{\infty} n \rho^n \quad \text{avec } \rho = \frac{\lambda}{\mu} \\ &= \rho (1 - \rho) \sum_{n=1}^{\infty} n \rho^{n-1} \\ &= \rho (1 - \rho) \left(\sum_{n=0}^{\infty} \rho^n\right)' \\ &= \rho (1 - \rho) \left[\frac{1}{(1 - \rho)^2}\right] \\ L &= \frac{\rho}{(1 - \rho)} \end{aligned}$$

b) Le nombre moyen de clients dans la file

$$\begin{aligned} L_q &= \sum_{n=0}^{\infty} (n - 1) p_n = \sum_{n=0}^{\infty} n p_n - \sum_{n=1}^{\infty} p_n \\ &= \sum_{n=0}^{\infty} n p_n - [\sum_{n=0}^{\infty} p_n - p_0] \\ &= \frac{\rho}{(1 - \rho)} - [1 - (1 - \rho)] \\ &= \frac{\rho - \rho(1 - \rho)}{(1 - \rho)} \\ L_q &= \frac{\rho^2}{(1 - \rho)} \end{aligned}$$

On peut obtenir maintenant le temps moyen de séjour et le temps moyen d'attente dans la file (W et W_q) soit d'après les formules de Little, soit directement à partir de la distribution stationnaire du système.

c) Le temps moyen de séjour dans le système

$$W = \frac{1}{\lambda} L = \frac{1}{\mu - \lambda}.$$

a) Le temps moyen d'attente dans la file

$$W_q = \frac{1}{\lambda} L_q = \frac{\lambda}{\mu(\mu - \lambda)}.$$

3.6. Système d'attente M/G/1

Ce système est formé d'une file FIFO à capacité illimitée, d'un seul serveur. Le processus des arrivées est supposé poissonien de taux λ , mais le temps de service est distribuée selon une variable aléatoire de loi générale X qui n'est plus exponentielle pouvant être uniforme, gamma, normale, constante,...

On est ainsi amené à considérer des système d'attente du type M/G/1. De nombreuses méthodes ont été développées pour les étudier; nous nous limiterons ci-après à la présentation d'une approche connue sous le nom de chaîne de Markov induite [3].

3.6.1. Chaîne de Markov induite

Soit donc un système d'attente du type M/G/1; le processus des arrivées est poissonien de paramètre λ et le temps de service Y est réparti suivant une loi quelconque $B(t)$. Pour des raisons de commodité, nous admettrons que Y est continu, sa densité sera notée $b(t)$ et son espérance mathématique $E(Y) = \frac{1}{\mu}$. Nous nous proposons de déterminer la distribution stationnaire et les caractéristiques usuelles du processus $\{X(t); t \geq 0\}$ où $X(t)$ est le nombre de clients dans le système à l'instant t .

A cet effet, nous considérons le processus $X(t)$ aux instant t_1, t_2, t_3, \dots où des clients terminent leur service et quittent le système. On définit ainsi un processus stochastique à temps discret : $\{X_n = X(t_n); n = 1, 2, \dots\}$

où t_n est l'instant juste après le départ du n -ième client. Pour vérifier que cette suite de variables aléatoires est une chaîne de Markov à temps discret, on considère le nombre A_n de clients qui entrent dans le système pendant que le n -ième client est servi [3].

Les variables aléatoires A_n sont indépendantes entre elles; leur distribution commune est:

$$P(A_n = k) = a_k = \int_0^{\infty} \frac{e^{-\lambda t} (\lambda t)^k}{k!} b(t) dt$$

où $a_k > 0$ ($k = 0, 1, 2, \dots$). Alors

$$X_{n+1} = \begin{cases} X_n - 1 + A_{n+1} & \text{si } X_n \geq 0 \\ A_{n+1} & \text{si } X_n = 0 \end{cases}$$

($n = 1, 2, \dots$), relation qui s'écrit également

$$X_{n+1} = X_n - \delta_n + A_{n+1}$$

$$\delta_n = \begin{cases} 1 & \text{si } X_n > 0 \\ 0 & \text{si } X_n = 0 \end{cases}$$

La suite de variables aléatoires $\{X(n); n \geq 1\}$ s'appelle chaîne de Markov induite [1] du processus $\{X(t); t \geq 0\}$. Ses probabilités de transition sont:

$$p_{ij} = P(X_{n+1} = j / X_n = i)$$

Se calculent par

$$p_{0j} = a_j \quad \text{si } j \geq 0$$

$$p_{ij} = a_{j-i+1} \quad \text{si } 1 \leq i \leq j+1$$

$$p_{ij} = 0 \quad \text{ailleurs,}$$

et la matrice des probabilités de transition prend la forme

$$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 & \dots \\ a_0 & a_1 & a_2 & a_3 & \dots \\ 0 & a_0 & a_1 & a_2 & \dots \\ 0 & 0 & a_0 & a_1 & \dots \\ \cdot & \cdot & \cdot & \cdot & \dots \end{pmatrix}$$

Puisque on peut passer de chaque état vers n'importe quel autre état, il s'agit d'une chaîne de

Markov irréductible dont on peut montrer quelle converge vers une distribution limite si $\rho = \frac{\lambda}{\mu} < 1$.

3.6.2. Régime stationnaire du système

Supposons que $\rho < 1$, et soit $\Pi = (\pi_1, \pi_2, \pi_3, \dots)$ la distribution stationnaire de la chaîne de Markov induite.

Il ne sera généralement pas possible de trouver la distribution Π elle-même, mais nous pouvons calculer la fonction génératrice correspondante $\Pi(z)$.

On a d'après la définition de la distribution de probabilité discrète stationnaire par rapport à une matrice stochastique P: $\Pi P = \Pi$

où
$$\pi_j = \sum_{i=0}^{\infty} p_{ij} \pi_i \quad (j = 0, 1, 2, \dots)$$

ce qui s'écrit également

$$\begin{aligned} \pi_j &= a_j \pi_0 + \sum_{i=1}^{j+1} a_{j-i+1} \pi_i \\ &= a_j \pi_0 + \sum_{i=0}^{j+1} a_{j-i+1} \pi_i - a_{j+1} \pi_0 \quad (j = 0, 1, 2, \dots) \end{aligned}$$

Si l'on multiplie cette équation par z^j et si l'on somme sur j, on a

$$\sum_{j=0}^{\infty} \pi_j z^j = \pi_0 \sum_{j=0}^{\infty} a_j z^j + \frac{1}{z} \sum_{j=0}^{\infty} c_{j+1} z^{j+1} - \frac{\pi_0}{z} \sum_{j=0}^{\infty} a_{j+1} z^{j+1}$$

où $c_{j+1} = \sum_{i=0}^{j+1} a_{j-i+1} \pi_i$ est le terme général du produit de convolution des deux distributions $A = (a_0, a_1, a_2, \dots)$ et Π .

En introduisant les fonctions génératrices:

$$\Pi(z) = \sum_{i=0}^{\infty} \pi_i z^i,$$

$$A(z) = \sum_{i=0}^{\infty} a_i z^i$$

et

$$C(z) = \sum_{j=0}^{\infty} c_j z^j = \Pi(z)A(z),$$

on obtient

$$\Pi(z) = \pi_0 A(z) + \frac{1}{z} [C(z) - c_0] - \frac{\pi_0}{z} [A(z) - a_0]$$

d'où

$$\Pi(z) = \frac{\pi_0 A(z)(z-1)}{z-A(z)}.$$

De la relation

$$X_{n+1} = X_n - \delta_n + A_{n+1}$$

on tire, puisqu'on se trouve en régime stationnaire,

$$E(A_{n+1}) = E(\delta_n) = P(\delta_n > 0) = P(X_n > 0) = 1 - P(X_n = 0).$$

D'où $\pi_0 = 1 - \rho$

et la fonction génératrice de la distribution stationnaire s'écrit

$$\Pi(z) = \frac{(1-\rho) A(z)(z-1)}{z-A(z)}$$

cette formule est appelée la formule de Pollaczek-Kinchin [3].

3.6.3. Caractéristiques du système M/G/1

a) Le nombre moyen de clients dans le système:

pour calculer le nombre moyen de clients dans le système en régime stationnaire,

$L = E(X)$, on pourrait s'appuyer sur la relation $E(X) = \lim_{z \rightarrow 1} \Pi'(z)$.

Ce calcul s'avérant plutôt pénible. Par contre, en utilisant la relation

$$X_{n+1} = X_n - \delta_n + A_{n+1}.$$

$$L = E(X_n) = \rho + \frac{\rho^2 + \lambda^2 \text{Var}(Y)}{2(1 - \rho)}$$

Où Y est la durée de service.

b) Le nombre moyen de clients dans la file est :

$$L_q = \frac{\rho^2 + \lambda^2 \text{Var}(Y)}{2(1 - \rho)}$$

c) Le temps moyen de séjour dans le système :

$$W = \frac{1}{\mu} + \lambda \frac{Var(Y) + \frac{1}{\mu^2}}{2(1 - \rho)}$$

d) Le temps moyen d'attente dans la file :

$$W_q = \lambda \frac{Var(Y) + \frac{1}{\mu^2}}{2(1 - \rho)}$$

3.7. Conclusion

Dans ce chapitre, nous avons rappelé et présenté les concepts et technique de base de la théorie de file d'attente classiques. Plus précisément, on a exposé quelques modèles d'attente particuliers et on a donné leurs principales caractéristiques, tout en abordant les files markoviennes et les files non markoviennes.

Chapitre - **IV**

Application & comparaison

4.1. Introduction

Générer une séquence de variable d'entrée suivant une loi donnée pour un modèle de simulation cela entraîne une génération de nombre aléatoire suivant une loi uniforme sur $[0,1]$ ce dernier peut se faire avec le logiciel MATLAB

Dans ce travail nous voulons faire cette génération avec les deux méthodes suivantes :

- ❖ hypercube latin
- ❖ descriptif améliorée

La génération en utilisent la méthode descriptif améliorée ce fait avec un générateur intègre ou MATLAB et le générateur de hypercube latin est disponible sous MATLAB.

Pour utiliser les deux méthodes nous proposons un problème de file d'attente $M / G / 1$, le flux des arrivées est Poissonien de paramètre λ , et la durée de service suit une loi générale. Nous ferons la simulation à l'aide de la méthode inversion.

Afin de comparer entre LHS et EDA on applique l'erreur absolue entre les résultats théorique et celui de la simulation , après on a appliquer aussi le critère de la variance pour les résultats obtenus dans le problème de file d'attente $M / G / 1$.

4.2. Présentation de MATLAB

MATLAB est un système interactif de programmation scientifique, pour le calcul numérique et la visualisation graphique, basé sur la représentation matricielle des données. Le nom dérive de cette représentation : MATLAB = MATrix LABoratory.

Matlab permet de faire des calculs mathématiques 'semi-formels'. Il connaît un grand nombre d'opérations ou de fonctions mathématiques : fonctions usuelles, calcul matriciel, fonctions plus spécifiques du signal, etc.

MATLAB étant disponible sur différentes plateformes matérielles, on accède au logiciel en lançant l'exécutable **matlab** suivant la procédure habituelle de l'environnement concerné. On se retrouve alors dans un environnement dit *fenêtre de commande* dans lequel on peut écrire des commandes ou exécuter des fonctions. La sortie du logiciel s'effectue en tapant **quit** ou **exit**. Sur les systèmes qui permettent le multi-fenêtrage, on peut travailler sous MATLAB dans une fenêtre et conserver son éditeur de texte dans une autre afin de modifier simplement les fonctions ou 'programmes matlab' à mettre au point.

4.3. Comment générer des variables avec la méthode LHS

Pour générer un échantillon X de LHS nous utilisons la fonction `lhsdesign(n,k)`

Tel que n est le nombre de valeur sur chacun des k variable.

Nous exécutons `lhsdesign` pour n=20 et k=1 comme suit :

```
>> x = lhsdesign(20,1)
```

x =

0.4703

0.0989

0.6287

0.1344

0.4419

0.5411

0.1789

0.3453

0.6701

0.2765

0.8152

0.9650

0.7181

0.3983

0.5966

0.0340

0.9235

0.8673

0.7796

0.2090

4.4. Comment générer des variables avec la méthode EDA

Pour générer une variable avec EDA nous ferons appel à une fonction quand on a programmée sous MTLAB qui s'appelle RDSN (), cette fonction est faite spécialement pour des variables à une dimension.

A travers cet exemple, nous allons voir les résultats obtenue par la fonction RDSN ()

Dès l'exécution du RDSN (), Matlab nous donne la main pour introduire la taille n de l'échantillon X.

```
>> X=RDSN()
```

On considère que $n=20$ et on obtient le résultat suivant :

X=

```
0.0366  0.0610  0.0854  0.1098  0.1341  0.1585  0.1829  0.2073  0.0122  0.2317
0.2561  0.2805  0.3049  0.3293  0.3537  0.3780  0.4024  0.4268  0.4512  0.4756
```

4.5. Problématique

Une clinique dispose d'un service de neurologie tenu par un seul médecin. Les malades se présentent selon un processus de poisson et la durée moyenne entre deux arrivées successives est $1/\lambda$ minute, et les durées des soins sont indépendantes et suivant une loi générale G pour chaque malade.

Les malades sont soignés dans le cabinet du médecin suivant leur ordre d'arrivée et il n'y a pas de limitation de place dans le service de neurologie.

Ce problème considère 5 variables de sortie ayant chacune un paramètre à estimer qui est la moyenne considères par les caractéristiques suivantes:

- Le temps moyen de séjour (Le temps d'attente dans le système) noté \bar{W} .
- Le temps moyen d'attente dans la file \bar{W}_q .

- a) Le nombre moyen de malades dans le système noté \bar{L} .
- b) Le nombre moyen de malades dans la file d'attente noté \bar{L}_q .
- c) Le pourcentage d'utilisation des places noté $\bar{\rho}$.

D'après l'énoncé on constate que on a affaire à un phénomène de file d'attente simple à un serveur

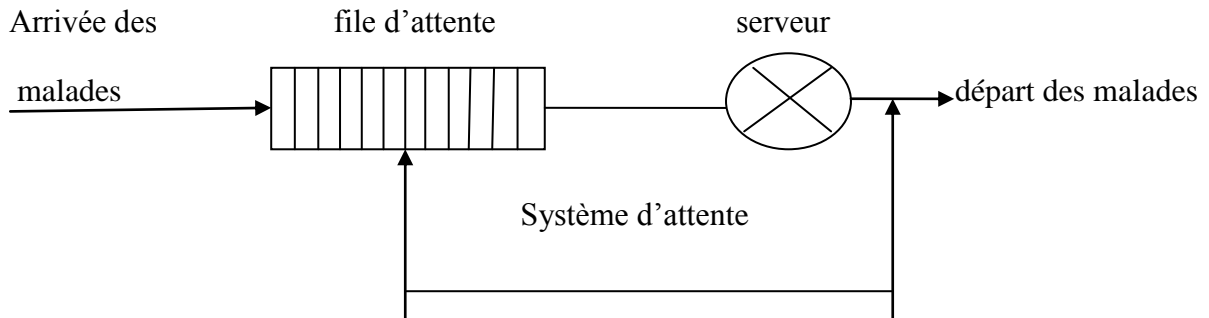


Figure 4.1. File d'attente simple à un serveur

4.6. Résolution avec la simulation

Dans la résolution par simulation à événements discrets nous utilisons l'approche par événements

C'est l'approche de base :

- Identification des différents types d'événements possibles au cours de la durée de vie du système.
- Description de la logique de fonctionnement entre événements : déterminer les changements d'états correspondant à chaque événement et les événements qui en résultent.
- L'utilisation de calendrier d'événements ou échéancier : liste des événements et leurs dates d'occurrence

Notre but est de simuler mais on fournit des échantillons avec les deux méthodes LHS et EDA.

Afin de réaliser notre objectif nous avons créé une fonction `mmain()` qui rend comme les résultats suivants:

Le nombre moyen de malades dans le système et dans la file d'attente, le temps moyen d'attente dans le système et le pourcentage d'utilisation des places.

mmain() a son tour fait appel à des fonctions qui produisent des échantillons LHS et EDA.

Pour pouvoir exécuter la fonction mmain() plusieurs fois ; on a crée une autre fonction nommer final() qui peut exécuter N histoires de mmain() , elle renvoie la moyenne et la variance de ses exécutions pour chacune des caractéristiques de la file M/G/1 .

4.7. Résultats de simulation pour différent lois générale

Lors de la simulation on considère N réplifications de simulation où chaque réplification considéré un nombre de malades = 1000 comme une période de simulation (temps d'arrêt) .

4.7.1. En utilisant le critère de la variance

Nous avons considéré le cas où le système n'est pas chargé ($\rho = 0.3$) et le cas où le système est chargé ($\rho = 0.9$) .

- **G suit une loi Uniforme [α, β]**

Deux expérience sont effectuées avec différent paramètres d'une loi Uniforme (α, β). Les paramètres de la loi Uniforme (α, β) sont choisis tels que $\alpha = 1$ et $\beta = 3$. Les valeurs théorique du problème étudiée sont $W = 2.4643$, $Wq = 0.4643$, $L = 0.3696$, $Lq = 0.0696$ et pour $\alpha = 0$ et $\beta = 2$; les valeurs théorique du problème étudiée sont $W = 7.00$, $Wq = 6.00$, $L = 6.30$, $Lq = 5.40$.

Pour le cas d'une loi Uniforme avec $\alpha = 1$, $\beta = 3$ et $\rho = 0.3$, le résumé des résultats de la file M/G/1 sont présentées dont les tableaux suivant:

Estimateur	LHS					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	0.4297	0.4019	0.3952	0.3957	0.4114	0.4061
Temps moyen dans le système	2.4262	2.4027	2.3898	2.3956	2.4121	2.4057
Nombre moyen dans la file	0.0703	0.0633	0.0620	0.0626	0.0656	0.0642
Nombre moyen dans le système	0.3958	0.3779	0.3743	0.3780	0.3838	0.3798
L'utilisation du serveur	0.3177	0.3146	0.3123	0.3154	0.3182	0.3156

Table 4.1. La moyenne des paramètres estimer par la simulation en utilisant LHS

Estimateur	EDA					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	0.4555	0.4750	0.4574	0.4554	0.4585	0.4588
Temps moyen dans le système	2.4496	2.4817	2.4542	2.4557	2.4569	2.4605
Nombre moyen dans la file	0.0684	0.0710	0.0688	0.0684	0.0686	0.0691
Nombre moyen dans le système	0.3676	0.3700	0.3684	0.3683	0.3666	0.3697
L'utilisation du serveur	0.2991	0.2990	0.2996	0.2999	0.2980	0.3006

Table 4.2. La moyenne des paramètres estimer par la simulation en utilisant EDA

Interprétation

Les résultats de la simulation par LHS et EDA qui sont données dans le tableau 4.1 et tableau 4.2 sont proche aux valeurs théoriques, mais avec EDA, on obtient des valeurs plus proche des valeurs théoriques.

Estimateur	LHS					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	0.0010	0.0049	0.0027	0.0021	0.0030	0.0024
Temps moyen dans le système	0.0012	0.0067	0.0031	0.0033	0.0040	0.0027
Nombre moyen dans la file	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001
Nombre moyen dans le système	0.0001	0.0004	0.0002	0.0002	0.0003	0.0003
L'utilisation du serveur	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001

Table 4.3. La variance des estimateurs obtenue par LHS

où $\rho = 0.3$

Estimateur	EDA					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	0.0016	0.0026	0.0011	0.0020	0.0017	0.0017
Temps moyen dans le système	0.0014	0.0035	0.0013	0.0015	0.0021	0.0021
Nombre moyen dans la file	0.0001	0.0001	0.0000	0.0001	0.0001	0.0001
Nombre moyen dans le système	0.0004	0.0002	0.0001	0.0002	0.0003	0.0002
L'utilisation du serveur	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001

Table 4.4. La variance des estimateurs obtenue par EDA

où $\rho = 0.3$

Interprétation

On remarque que les variances calculées dans les tableaux 4.3 et 4.4 soit avec la méthode de LHS ou EDA sont presque proches, mais celle de EDA sont plus petite que LHS alors on peut dire que la meilleure méthode pour réduire la variance est celle de EDA.

Les résultats obtenues de M/G/1 où G suit une loi Uniforme avec les paramètres $\alpha = 0$, $\beta = 2$ et $\rho = 0.9$ sont présentés dans les tableaux suivant:

Estimateur	LHS					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	6.9513	7.3028	6.8930	6.6704	7.9797	7.6254
Temps moyen dans le système	7.9516	8.2936	7.8911	7.6643	7.9776	8.6316
Nombre moyen dans la file	6.5975	7.0188	6.5262	6.3943	6.5824	7.2228
Nombre moyen dans le système	7.5354	7.9563	7.4619	7.3394	7.5242	8.1409
L'utilisation du serveur	0.9180	0.9375	0.9357	0.9451	0.9218	0.9180

Table 4.5. La moyenne des paramètres estimer par la simulation en utilisant LHS

où $\rho = 0.9$

Estimateur	EDA					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	4.9212	5.9690	4.9105	5.4013	5.2285	5.2891
Temps moyen dans le système	5.9121	6.9663	5.9083	6.4028	6.2303	6.2887
Nombre moyen dans la file	4.4946	5.4914	4.4458	4.9153	4.7119	4.7753
Nombre moyen dans le système	5.3878	6.3951	5.3381	5.8130	5.6070	5.6686
L'utilisation du serveur	0.8932	0.9037	0.8924	0.8978	0.8951	0.8933

Table 4.6. La moyenne des paramètres estimer par la simulation en utilisant EDA où $\rho = 0.9$

Interprétation

Les résultats de la simulation par LHS et EDA qui sont données dans le tableau 4.5 et tableau 4.6 sont proche aux valeurs théoriques, mais avec EDA on obtient des valeurs plus près des valeurs théoriques.

Estimateur	LHS					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	1.4440	2.3573	2.1627	4.3930	2.3195	4.2511
Temps moyen dans le système	1.4723	2.3871	2.1864	4.4102	2.3426	4.2783
Nombre moyen dans la file	1.2263	2.1248	1.9939	4.3245	2.0804	3.7537
Nombre moyen dans le système	1.2611	2.1703	2.0394	4.4305	2.1396	3.8241
L'utilisation du serveur	0.0004	0.0005	0.0005	0.0010	0.0008	0.0006

Table 4.7. La variance des estimateurs obtenue par LHS

où $\rho = 0.9$

Estimateur	EDA					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	1.3247	1.1120	1.1609	1.3247	1.5436	4.6487
Temps moyen dans le système	1.3210	1.1103	2.2026	1.3210	1.5893	4.5834
Nombre moyen dans la file	1.3368	1.2679	1.7514	1.3368	1.6402	4.6308
Nombre moyen dans le système	1.3497	1.3465	1.8324	1.3497	2.7053	4.6156
L'utilisation du serveur	0.00001	0.0002	0.0001	0.0001	0.0002	0.0000

Table 4.8. La variance des estimateurs obtenue par EDA

où $\rho = 0.9$

Interprétation

On remarque que les variances des estimateurs dans les tableaux 4.7 et 4.8 soit avec la méthode de LHS ou EDA sont presque proche, mais celle de EDA sont plus petite que LHS, alors on peut dire que la meilleure méthode pour réduire la variance est celle de EDA.

- **G suit une loi Gamma (α, β)**

Deux expérience sont effectuées avec différent paramètres d'une loi de Gamma (α, β). Les paramètres de la loi Gamma (α, β) sont choisis tels que $(\alpha, \beta) = (2,5)$ pour une faible utilisation et $(\alpha, \beta) = (3,3)$ pour un système chargé. les valeurs théorique pour le premier cas sont $Wq = 0.1286$, $W = 0.5286$, $Lq = 0.0964$, $L = 0.3964$, et pour le deuxième cas les valeurs théorique sont $Wq = 6.00$, $W = 7.00$, $Lq = 5.40$, $L = 6.30$.

Le résumé des résultats de la file M/G/1 est présentée dont les tableaux suivant:

Pour une loi de gamma de paramètre $\alpha = 2$, $\beta = 5$ et $\rho = 0.3$.

Estimateur	LHS					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	0.0997	0.0959	0.0937	0.0972	0.0943	0.0957
Temps moyen dans le système	0.4786	0.4760	0.4748	0.4755	0.4721	0.4755
Nombre moyen dans la file	0.0805	0.0760	0.0740	0.0772	0.0748	0.0758
Nombre moyen dans le système	0.3857	0.3768	0.3740	0.3773	0.3738	0.3762
L'utilisation du serveur	0.3052	0.3008	0.3000	0.3001	0.2990	0.3003

Table 4.9. Les résultats de simulation d'une file M/G/1 par LHS

Estimateur	EDA					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	0.1208	0.1255	0.1304	0.1262	0.1290	0.1273
Temps moyen dans le système	0.5174	0.5262	0.5327	0.5251	0.5283	0.5273
Nombre moyen dans la file	0.0902	0.0939	0.0982	0.0947	0.0975	0.0956
Nombre moyen dans le système	0.3857	0.3936	0.4004	0.3930	0.3985	0.3952
L'utilisation du serveur	0.2955	0.2998	0.3022	0.2983	0.3010	0.2996

Table 4.10. Les résultats de simulation d'une file M/G/1 par EDA

Interprétation

Les résultats de la simulation par LHS et EDA qui sont données dans le tableau 4.9 et tableau 4.10 sont proche aux valeurs théoriques, mais avec EDA, on obtient des valeurs plus près du valeurs théoriques.

Estimateur	LHS					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	0.0001705	0.0002937	0.0002315	0.0002153	0.0002036	0.0002336
Temps moyen dans le système	0.0002631	0.0004969	0.0003538	0.0004345	0.0003237	0.0003858
Nombre moyen dans la file	0.0001175	0.0001592	0.0001711	0.0001626	0.0001521	0.0001704
Nombre moyen dans le système	0.0002790	0.0003773	0.0004824	0.0005210	0.0004792	0.0005114
L'utilisation du serveur	0.0000634	0.0001292	0.0001179	0.0001339	0.0001489	0.0001367

Table 4.11. La variance des estimateurs obtenue par LHS

où $\rho = 0.3$

Estimateur	EDA					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	0.0001577	0.0000853	0.0001187	0.0001548	0.0001399	0.0001429
Temps moyen dans le système	0.0002320	0.0002062	0.0002510	0.0002891	0.0002618	0.0002836
Nombre moyen dans la file	0.0001231	0.0000654	0.0001003	0.0001099	0.0000998	0.0001124
Nombre moyen dans le système	0.0002558	0.0002482	0.0004590	0.0003084	0.0002782	0.0004103
L'utilisation du serveur	0.0000489	0.0000788	0.0001804	0.0000912	0.0000835	0.0001319

Table 4.12. La variance des estimateurs obtenue par EDA

où $\rho = 0.3$

Interprétation

On constate que les variances calculées dans les tableaux 4.11 et 4.12 soit avec la méthode de LHS ou EDA sont presque proche, alors on peut dire que la meilleure méthode pour réduire la variance est celle de EDA.

Pour une loi de gamma de paramètre $\alpha = 3$, $\beta = 3$ et $\rho = 0.9$.

Estimateur	LHS					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	5.6403	5.1864	5.8473	5.9235	5.4623	5.2296
Temps moyen dans le système	5.5945	6.1354	5.7930	5.8693	5.4078	6.1763
Nombre moyen dans la file	4.4423	5.0238	4.6381	4.7337	5.2969	5.0385
Nombre moyen dans le système	5.3347	5.9216	5.5232	5.6179	5.1956	5.9335
L'utilisation du serveur	0.8924	0.8978	0.8851	0.8842	0.8987	0.8950

Table 4.13. Les résultats de simulation d'une file M/G/1 par LHS

où $\rho = 0.9$

Estimateur	EDA					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	5.7545	5.1607	5.5882	4.9687	5.2793	5.4295
Temps moyen dans le système	6.7595	6.1570	6.5890	5.9670	6.2783	6.4308
Nombre moyen dans la file	5.1564	4.6904	5.1039	4.5021	4.7923	4.9466
Nombre moyen dans le système	6.0410	5.5836	5.9979	5.3943	5.6832	5.8441
L'utilisation du serveur	0.8847	0.8932	0.8941	0.8922	0.8909	0.8975

Table 4.14. Les résultats de simulation d'une file M/G/1 par EDA où $\rho = 0.9$

Interprétation

Les résultats de la simulation par LHS et EDA qui sont données dans le tableau 4.9 et tableau 4.10 sont proche aux valeurs théoriques, mais avec EDA on obtient des valeurs plus près du valeurs théoriques.

Estimateur	LHS					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	3.5571	6.9462	4.5212	3.1436	7.2654	4.9348
Temps moyen dans le système	3.5547	6.9821	4.5458	3.1717	7.3005	4.9587
Nombre moyen dans la file	3.1605	6.3314	4.5847	2.9023	6.8733	4.7173
Nombre moyen dans le système	3.2195	6.4456	4.7051	3.0014	6.9613	4.8184
L'utilisation du serveur	0.0008	0.0010	0.0012	0.0012	0.0006	0.0009

Table 4.15. La variance des paramètres obtenue par LHS

où $\rho = 0.9$

Estimateur	EDA					
	N=10	N=20	N=30	N=40	N=50	N=100
Temps moyen dans la file	0.7169	2.2491	1.4331	3.1714	3.5913	2.9151
Temps moyen dans le système	0.7347	2.2543	1.4437	3.1963	3.6230	2.9469
Nombre moyen dans la file	0.7129	2.5599	1.4426	3.3844	3.8113	2.8656
Nombre moyen dans le système	0.7471	2.6417	1.4853	3.4817	3.9129	2.9335
L'utilisation du serveur	0.0005	0.0008	0.0006	0.0011	0.0010	0.0007

Tab 4.16. La variance des paramètres obtenue par EDA

où $\rho = 0.9$

Interprétation

On constate que les variances calculées dans les tableaux 4.11 et 4.12 soit avec la méthode de LHS ou EDA ne sont pas proche, alors déduit que la meilleure méthode pour réduire la variance est celle de EDA.

4.7.2. En utilisant le critère de la valeur absolue

Nous considérons les différents lois de service considérés avec le critère de la variance et on utilise le critère de la valeur absolue pour comparer nos estimateurs.

Les formules utilisées sont les suivantes:

$$W^* = |\bar{W} - W|, W_q^* = |\bar{W}_q - W_q|, L^* = |\bar{L} - L|, L_q^* = |\bar{L}_q - l_q|, \rho^* = |\bar{\rho} - \rho|.$$

G suit la loi	Erreur absolue	N=10	N=20	N=30	N=40	N=50	N=100
U (1,3)	W_q^*	0.0088	0.0107	0.0069	0.0089	0.0058	0.0055
	W^*	0.0147	0.0174	0.0101	0.0086	0.0074	0.0038
	L_q^*	0.0012	0.0014	0.0008	0.0012	0.0010	0.0005
	L^*	0.0020	0.0004	0.0012	0.0013	0.0030	0.0001
	ρ^*	0.0009	0.0010	0.0004	0.0001	0.0020	0.0006
U(0,2)	W_q^*	1.0788	0.0310	1.0895	0.5987	0.7715	0.7109
	W^*	1.0879	0.0337	1.0917	0.5972	0.7697	0.7113
	L_q^*	0.9054	0.0914	0.9542	0.4847	0.6881	0.6247
	L^*	0.9122	0.0951	0.9619	0.4870	0.6930	0.6314
	ρ^*	0.0068	0.0037	0.0076	0.0022	0.0049	0.0067
Gamma (2,5)	W_q^*	0.0078	0.0031	0.0018	0.0024	0.0004	0.0013
	W^*	0.0112	0.0024	0.0041	0.0035	0.0003	0.0013
	L_q^*	0.0062	0.0025	0.0018	0.0017	0.0011	0.0008
	L^*	0.0107	0.0028	0.0040	0.0034	0.0021	0.0012
	ρ^*	0.0009	0.0010	0.0004	0.0001	0.0020	0.0006
Gamma (3,3)	W_q^*	0.2455	0.8393	0.4118	1.0313	0.7207	0.5705
	W^*	0.2405	0.8430	0.4110	1.0330	0.7217	0.5692
	L_q^*	0.2436	0.7096	0.2961	0.8979	0.6077	0.4534
	L^*	0.2590	0.7164	0.3021	0.9057	0.6168	0.4559
	ρ^*	0.0153	0.0068	0.0059	0.0078	0.0091	0.0025

Table 4.17. Les valeurs empiriques des erreurs absolues obtenues par EDA

G suit la loi	Erreur absolue	N=10	N=20	N=30	N=40	N=50	N=60
U (1,3)	W_q^*	0.0346	0.0624	0.0691	0.0686	0.0529	0.0582
	W^*	0.0381	0.0616	0.0745	0.0687	0.0522	0.0586
	Lq	0.0007	0.0063	0.0076	0.0070	0.0040	0.0054
	L^*	0.0262	0.0083	0.0047	0.0084	0.0142	0.0102
	ρ^*	0.0177	0.0146	0.0123	0.0154	0.0182	0.0156
U(0,2)	W_q^*	0.9513	1.3028	0.8930	0.6704	1.9797	1.6254
	W^*	0.9516	1.2936	0.8911	0.6643	0.9776	1.6316
	Lq	1.1975	1.6188	1.1262	0.9943	1.1824	1.8228
	L^*	1.2354	1.6563	1.1619	1.0394	1.2242	1.8409
	ρ^*	0.0180	0.0375	0.0357	0.0451	0.0218	0.0180
Gamma (2,5)	W_q^*	0.0289	0.0327	0.0349	0.0314	0.0343	0.0329
	W^*	0.0500	0.0526	0.0538	0.0531	0.0565	0.0531
	Lq	0.0159	0.0204	0.0224	0.0192	0.0216	0.0206
	L^*	0.0107	0.0196	0.0224	0.0191	0.0226	0.0202
	ρ^*	0.0052	0.0008	0	0.0001	0.0010	0.0003
Gamma (3,3)	W_q^*	0.3597	0.8136	0.1527	0.0765	0.5377	0.7704
	W^*	1.4055	0.8646	1.2070	1.1307	1.5922	0.8237
	Lq	0.9577	0.3762	0.7619	0.6663	0.1031	0.3615
	L^*	0.9653	0.3784	0.7768	0.6821	1.1044	0.3665
	ρ^*	0.0076	0.0022	0.0149	0.0158	0.0013	0.0050

Table 4.18. Les valeurs empiriques des erreurs absolues obtenue par LHS

Interprétation

On remarque que les erreurs absolues dans les tableaux 4.17 et 4.18 sont proches de zéro, mais avec EDA on obtient des valeurs plus petite que LHS.

Conclusion général

L'objectif du travail présenté dans ce mémoire est l'étude et l'analyse du comportement des file d'attente M/G/1 théoriquement et par simulation.

Pour simuler nous avons utilisé le générateur LHS sous MATLAB et amélioré le générateur RDS sous MATLAB en intégrant le nombre d'histoire et le nombre de réplifications pour pouvoir l'utiliser dans notre application.

Dans notre application, nous avons comparé les deux méthodes d'échantillonnage LHS et EDA en appliquant deux critères celui de la variance et celui de l'erreur absolue. Les distributions du temps de service du système de file d'attente examinés sont la loi uniforme et la loi gamma. Nous avons considéré le cas où le système est chargé, $\rho = 0.9$, et le cas où le système est fluide (non chargé), $\rho = 0.3$.

Dans tous les tableaux, nous remarquons que la variance dépend du taux d'utilisation du système ρ . Lorsque ρ augmente la variance augmente et lorsque ρ diminue la variance diminue. Nous remarquons que la variance est réduite dans les différents cas étudiés.

On constate aussi que lorsque le système est chargé, l'utilisation du serveur est de l'ordre de 90% en général, donc le serveur est utilisé à plein temps, il ya donc, un encombrement dans le système. Par conséquent, on suggère de rajouter d'autre serveurs pour améliorer et diminuer le taux l'utilisation du serveur. Par ailleurs, quand le système n'est pas chargé, on a donc pas besoin d'ajouter d'autre serveurs.

Vu les résultats des variances et des erreurs absolues obtenus à travers la simulation de la file d'attente M/G/1, on conclue que la méthode EDA est meilleure que LHS, soit lorsque G suit la loi Uniforme ou la loi Gamma.

PERSPECTIVES

- Comparer EDA avec d'autres méthodes d'échantillonnage, par exemple les méthode de QMC
- Etudier d'autre loi de G pour notre système de file d'attente considéré, par exemple, Weibull.
- Appliquer les méthodes d'échantillonnage étudiées sur d'autres cas.

Bibliographie

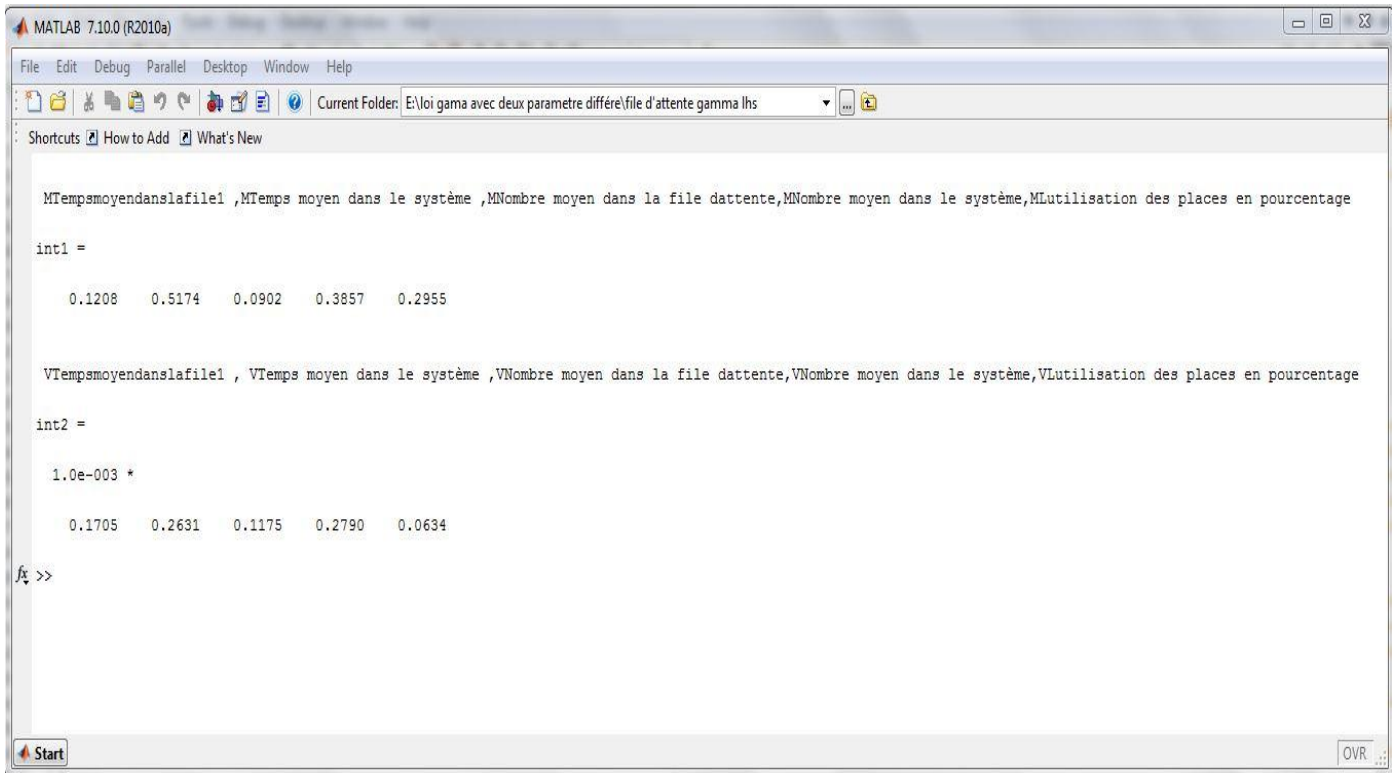
- [1] A. ALOUI et M. OURBIH-TARI, The use of refined descriptive sampling and applications in parallel Monte Carlo simulation, Computing and informatics, Vol.30, 2011.
- [2] A. Aloui , A. Zioui ,M. Ourbih-Tari , A general purpose module using refined descriptive sampling for installation in simulation systems, Comput stat DOI 10.1007/s00180-014-0545-7, 2014.
- [3] A. RUEGG, Processus stochastiques avec application aux phénomènes de file d'attente et de fiabilité, 1989.
- [4] A. Ricochon & N. Dantec , Mise en place d'un protocole d'évaluation de différentes méthodes d'échantillonnage Tome1 mars 2009.
- [5] C. Vassard, Autour des générateurs pseudo-aléatoires - Univers TI-Nspire,2011.
- [6] H. IMMEDIATO Cours de Statistiques (1^{ème} partie) MASS 42, Université Claude Bernard - Lyon 1, 2001.
- [7] L. ELIE et B. LAPERYRE, Introduction aux méthodes de Monte Carlo, septembre 2001.
- [8] L. Ourbih- Baghdali, M. Ourbih-Tari and A. Dahmani, A Pseudo Random Number Generator under Windows Using Refined Descriptive Sampling, Computer technology and application 4, 2013.
- [9] M. OURBIH-TARI et A. ALOUI Journal of Statistics: Advances in Theory and Applications Volume 1, Number 2, 2009.
- [10] M. Ourbih, Cours de Master 1 SAD " Méthodes de Monte Carlo", université de Bejaia, 2014.
- [11] M. Ourbih, Cours de 3^{ème} année STID "Simulation a événement discret" , Département Mathématique, université de béjaia, 2012.

- [12] M. TARI et A. DAHMANI Descriptive Sampling Improved Laboratory of Applied Mathematics Department of Mathematics University of Bejaia Algeria, pp 2.
- [13] M. TARI , A. DAHMANI Flowshop Simulator Using Different Sampling Methods ,International Journal Vol.5,No.2(2005).
- [14] P-J. Erard,P.Déguénon. Simulation par événements discrets : Concepts et réalisations en simula, Ada et Smalltalk ,1996.
- [15] R. Faivre, B. Iooss, S. Mahévas, D. Makowski, H. Monod, Analyse de sensibilité et exploration de modèles: Application aux sciences de la nature et de l'environnement. Edition 2013.

Annexe

Annexe A

L'exécution du programme final() pour une loi Gamma (2,5) avec N=10 donne :



The image shows a MATLAB 7.10.0 (R2010a) command window. The current folder is 'E:\loi gama avec deux parametre differe\file d'attente gamma lhs'. The command window displays the following output:

```
MTempsmoyendanslafile1 , MTemps moyen dans le système , MNombre moyen dans la file d'attente, MNombre moyen dans le système, MLutilisation des places en pourcentage  
int1 =  
    0.1208    0.5174    0.0902    0.3857    0.2955  
  
VTempsmoyendanslafile1 , VTemps moyen dans le système , VNombre moyen dans la file d'attente, VNombre moyen dans le système, VLutilisation des places en pourcentage  
int2 =  
1.0e-003 *  
    0.1705    0.2631    0.1175    0.2790    0.0634  
fx >>
```