République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaïa
Faculté des Sciences Exactes
Département Mathématiques

Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master en Mathématiques
Option : Probabilités Statistique et Applications

Thème

Application de l'échantillonnage descriptif amélioré dans les files d'attente

Réalisé par :

M^{elle} TIGHLIT Katia

Devant le Jury composé de :

M^{me} BOURAINE Louiza	Professeur	Présidente	Université A.Mira de Béjaia
M^{r} ZIOUI Arezki	M.C.B	Promoteur	Université A.Mira de Béjaia
M^{me} Idjis Khelidja	M.C.B	Examinatrice	Université A.Mira de Béjaia

Année universitaire : 2020 / 2021

Dédicaces

Je dédie ce modeste travail :

A ma chère mère,
A mon cher père,

Pour tout leurs sacrifices, leurs amours, leurs tendresses, et qui n'ont jamais cessé de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

A mes chères soeurs, pour leurs appuis et leurs encouragements.

A toute ma famille et ami(e)s pour leurs soutiens tout au long de mon parcours, universitére.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

K. Tighlit

Remerciements

Au premier lieu et avant tout je tiens à remercier Dieu qui m'a donné la force et la patience d'accomplir ce modeste travail.
Je tiens à exprimer aussi toute ma gratitude à mon encadreur Mr **A. Zioui** qui m'a offert la possibilité de réaliser ce travail sous sa direction, je le remercie également pour ses discussions profitables, ses encouragements,son entiére disponibilité, ses conseils judicieux, sa compréhension et ses suggestions.
J'adresse mes profonds remerciements aux membres de jury qui ont bien voulu évaluer et examiner mon travail : Mme **L. Bouraine** qui ma fait l'honneur de présider mon jury de soutenance et **Kh. Idjis** qui m'a fait l'honneur d'éxaminer mon travail.
Merci également à ma famille et en particulier mes parents, pour l'encouragement constant et leurs aides moraux sans retenue, et pour leurs soutiens inconditionnels tout au long de ces années d'études.
Enfin, mes remerciement vont à tous ceux qui m'ont soutenus ou qui, d'une maniére ou d'une autre, ont contribués à la réalisation de ce travail.

Table des matières

Table des figures

Liste des tableaux

Notations

X	Variable aléatoire ou vecteur de variable aléatoire.
x	Réalisation de la variable aléatoire X.
Y	Variable aléatoire ou vecteur de variable aléatoire.
y	Réalisation de la variable aléatoire Y.
U	Variable aléatoire distribuée selon $\mathcal{U}([0,1])$.
u	Réalisation de la variable aléatoire U.
F	Distribution cible de la densité f.
f	Densité de probabilité.
$\mathcal{U}([0,1])$	Distribution uniforme définie sur l'intervalle unité.
$\mathcal{N}(0,1)$	Distribution gaussienne centrée réduite.
$\varepsilon(\lambda)$	Distribution exponentielle de paramètre λ.
$\mathcal{L}(.)$	Loi d'une variable aléatoire.
σ_A	Fonction indicatrice qui vaut 1 sur l'ensemble A et 0 ailleurs.
n	Nombre total des itérations.
i	Indice des itérations d'un algorithme.
$i,\ j$	Indices généraux.
$(X(t))_{t\geq 0}$	Processus de Poisson.

$\mathbb{R}$	Ensemble des nombres réels.
$\mathbb{N}$	Ensemble des entiers naturels.
$\mathbb{N}^*$	Ensemble des entiers naturels non nuls.
$\mathbb{R}^d$	Ensemble des nombres réels de dimension d.
log	Logarithme népérien.
$\sim$	Suivre la loi, de loi....
v.a	Variable aléatoire.
i.i.d	Indépendante identiquement distribuée.
p.s	Presque sûrement.
TCL	Théorème centrale limite.
Var	Variance.
DS ou ED	Échantillonnage déscrptif
RDS ou EDA	'Echantillonnage déscrptif amélioré
MC	Monte-Carlo.
S.F.A	système File d'attente
W	Le temps moyen d'attente dans le système
W_q	Le temps moyen d'attente dans la fille
L	Le nombre moyen de clients dans le système
L_q	Le nombre moyen de clients en attente

Introduction Générale

L'échantillonnage est l'opération de générer un ensemble de valeur selon une loi de probabilité. Ces valeurs doivent vérifier un certain nombre de critéres : uniformité, indépendance,..., et quelques propriétés statistiques : la moyenne, la variance ,..., ces valeurs constituent les paramètres d'entrée du simulateur. Actuellement l'échantillonnage aléatoire (EA) est la méthode la plus utilisée dans la simulation de Monte-Carlo [14, 4, 7]. Dans cette procédure, les valeurs de l'échantillon sont générées en utilisant les générateurs de nombres aléatoires et la fonction de répartition inverse ou des méthodes équivalentes. Les estimateurs obtenus par EA varient entre les differentes histoires de la simulation, donc les estimateurs de la simulation dépendent de la procédure de l'EA.

La méthode de Monte-Carlo génére deux types d'erreurs, celui dû aux valeurs de l'échantillon et celui lié à la suite de ces veleurs. Pour remédier à ces deux inconvénients, plusieurs méthodes sont apparues, par exemple les méthodes quasi Monte-Carlo [17, 5, 9, 8], l'échantillonnage descriptif [16], l'échantillonnage descriptif amélioré [1, 2, 10, 11].

La méthode d'échantillonnage descriptif amélioré (EDA) basée sur le principe de l'ED élimine les inconvénients de l'ED et garde ses avantages. L'EDA est définie par un bloc de sous-ensembles de nombres réguliers dont les tailles sont des nombres premiers. Ces derniers sont choisis aléatoirement. Ce bloc doit étre situé à l'intérieur d'un générateur distribuant ces nombres réguliers à la demande de la simulation. La procédure s'arrête lorsque la simulation se termine.

Ce mémoire est organisé sous forme de 4 chapitres :

- Le premier chapitre couvre les concepts de base de simulation à événement discret, les nombres aléatoires et pseudo-aléatoires et la génération d'échantillon suivant differentes lois de probabilités.
- Dans le deuxième chapitre, nous présentons les différentes méthodes d'échantillonnages à savoir la méthode de Monte-Carlo,l'échantillonnage descriptif et l'échantillonnage descriptif amélioré,et nous présentons ainsi les avantages et les inconvénients de chacune.
- Le chapitre trois, couvre un rappel sur le processus de naissance et de mort et le système markoviens de files d'attente M/M/1.

- Le chapitre quatre est consacré à l'application de la méthode d'échantillonnage descriptif amélioré au système de files d'attente M/M/1 suivie d'une discussion.

Chapitre 1

Simulation des variables aléatoires

1.1 Introduction

Trouver une solution analytique d'un modèle probabiliste est souvent impossible. Dans ce cas, la seule manière d'étudier le système est donc la simulation. Simuler un système avec des paramètres ou des conditions initiales probabilistes demande la capacité de générer des nombres selon une distribution de probabilités. Dans ce chapitre, nous abordons la notion de la simulation à événements discrets et nous donnons quelques techniques de simulation permettant de générer un échantillon de variables aléatoires distribuées selon une densité donnée.

1.2 Simulation à événements discrets

La simulation à événements discrets est une technique utilisée dans le cadre de l'étude de la dynamique des systèmes, elle consiste en une modélisation informatique où le changement de l'état d'un système, au cours du temps, est une suite d'événements discrets. Chaque événement arrive à un instant donné et modifie l'état du système.

1.2.1 Présentation du problème de simulation

Les techniques de simulation Monte Carlo sont utilisées pour simuler des systèmes déterministes avec des paramètres ou des entrées stochastiques. La simulation est l'un des outils d'aide à la décision les plus efficaces à la disposition des concepteurs et des gestionnaires des systèmes complexes. Elle consiste à construire un modèle d'un système réel et à conduire des expériences sur ce modèle afin de comprendre le comportement de ce système et d'en améliorer ses performances.

1.2.2 Caractéristiques des modèles de simulation

• Un système

Un système peut-être vu comme un ensemble d'éléments qui agissent et interagissent dans un objectif commun.

• Un modèle

Un modèle est une représentation simplifieé de la réalité.

• Un processus

Un processus est une suite continue de faits, de phénomènes présentant une certaine unité ou une certaine régularité dans leur déroulement

• Une activité

Une activité est un intervalle de temps pendant lequel l'état de la ressource ne change pas.

• Une entité

une entité est une chose concrète ou abstraite comprenant des associations entre ces choses, par exemple une personne, un objet, une idée, un processus, etc.

• Un événement

Un événement est un ensemble des faits qui créent un changement d'état de la ressource.

• Un simulateur

Pour finir, un similateur est un programme contenant l'algorithme utilisé pour simuler le système étudié. Il est constitué d'un ensemble d'entités qui décrivent une composante du système réel.

1.2.3 Classification des modèles de simulation

les modèles de simulation peuvent-être classés en deux catégories :

1. Modèles déterministes.
2. Modèles stochastiques.

(*) Un modèle de simulation déterministe est un modèle qui ne fait pas intervenir le hasard. Comme exemple, un modèle, dans lequel on considère, que les clients arrivent à intervalles de temps réguliers, toutes les 5 minutes, et le temps de traitement d'un client par le postier est de 10 minutes.

(*) Un modèle de simulation stochastique est tout le contraire d'un modèle déterministe, le temps d'arrivée et le temps de traitement obéissent à des lois de probabilités. Par exemple : le temps d'arrivée $\sim\ \varepsilon(\frac{1}{2})$ et le temps de traitement $\sim\ \varepsilon(\frac{1}{4})$

Exemple 1.1. *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 (clients, voitures...) à un nombre limité de ressources (les guichets d'une poste, une station service). On appelle file d'attente l'ensemble des clients qui attendent d'être servis y compris celui qui se fait servir.*

1.2.4 Approche de simulation et modèle

Comme cela a été brièvement vu précédemment, la simulation à événements discrets consiste à reproduire, événement par événement, l'évolution d'un système au cours du temps. Il existe deux approches :

1. Approche par événements.
2. Approche par processus.

1.2.4.1 Approche par événements

Cette approche demande la définition d'une liste qui contient l'ensemble des événements futurs à venir. Cette approche est implémentée par une boucle qui execute les instructions suivantes (voir figure (1.1)) :

1. Gestion de l'événement actuel.
2. Exécute la liste des modifications liées à l'événements.
3. mise à jour la liste avec des événements futurs

Cette approche peut être implémentée par le langage de programmation MATLAB où il est possible de définir une liste. Toutefois, s'ils existent plusieurs types d'événements, l'implémentation pourrait être rendue plus aisée par l'existence d'une structure de données du genre file d'attente ou arbre.

1.2.4.2 Approche par processus

Cette approche associe un processus à chaque entité et modélise le système comme un ensemble d'entités interagissant (voir figure (1.2)). Dans cette approche :

1. Chaque processus modélise l'avancement du temps.
2. Un exemple de processus est le processus client qui arrive, se met en attente, est réveillée par la mise à disposition d'un guichet, attend le temps d'exécution de l'opération et puis s'en va du système.
3. Le paradigme orienté-objet est souvent utilisé pour modéliser une représentation process-based du système.

Cette approche peut être implémentée par les langages orienté objet : C++, Java, Delphi....

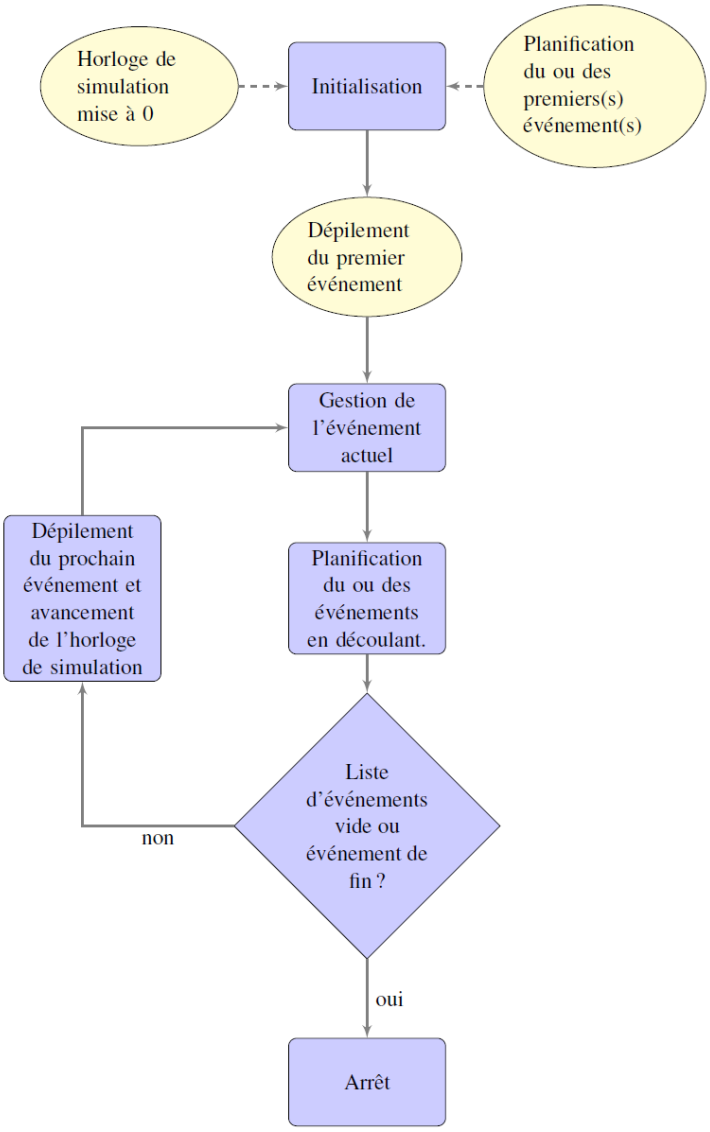


FIGURE 1.1 – Diagramme de l'approche par événement.

1.2.5 Les étapes de simulation

La conduite d'une étude de simulation comprend trois étapes principales : l'analyse du problème, la construction du modèle et l'exploitation de ce modèle.

- **Etape 1 : Analyse du problème**

  Cette étape, permet de préciser le contexte de l'étude et elle se décompose en :
  — Identification du problème ; spécification des objectifs.
  — Réalisation d'une première ébauche du modèle qui a pour but d'en délimiter les frontières et de spécifier les données dont on a besoin.

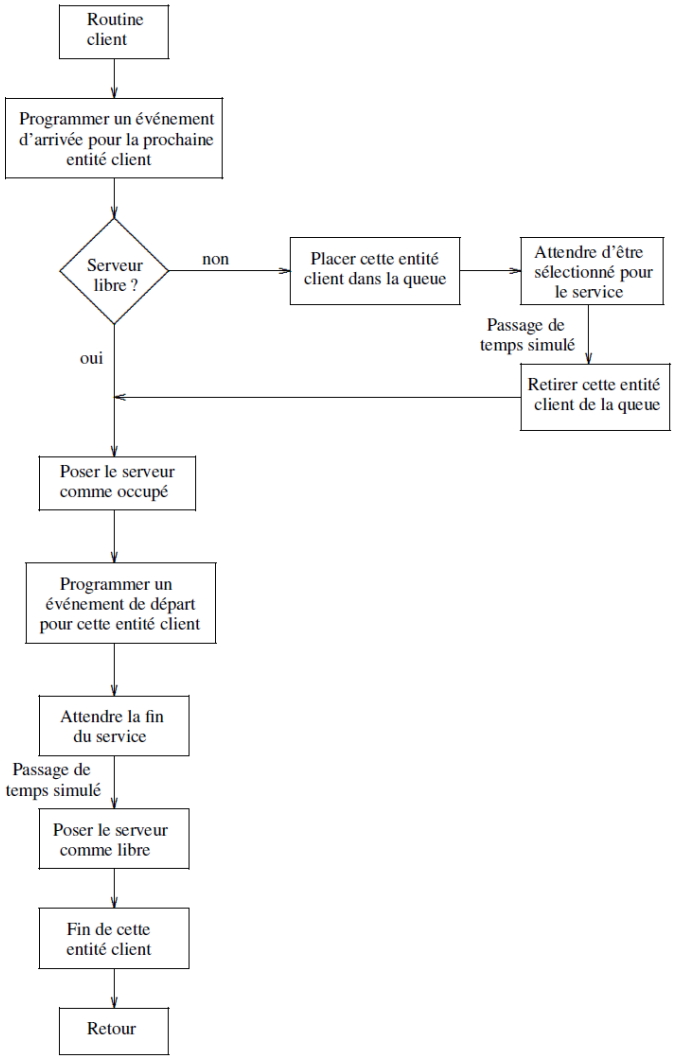


FIGURE 1.2 – Diagramme de l'approche par processus.

— Validation auprès de l'utilisateur (celui qui est à l'origine de l'étude).

Le but à atteindre dans cette étape est de construire un modèle valide qui soit le plus simple possible, tout en restant cohérent avec les objectifs de l'étude

- **Etape 2 : Construction du modèle**

Elle comprend la modélisation logico-mathématique qui peut être facilitée par un outil graphique, et la programmation proprement dite. Il est important dès cette étape de construire un programme facilement modifiable et en particulier de distinguer clairement le système physique, le système de conduite et le système d'information. Dans sa première itération, cette étape se termine par une validation qui consiste à comparer

le comportement du modèle avec celui du système physique qu'il est censé représenter.

- **Etape 3 : Exploitation du modèle**

Quand le modèle est validé, il peut servir à l'évaluation du comportement dynamique du système. Cette phase nécessite une définition précise de la campagne d'exploitation (quelles hypothèses veut-on vérifier, dans quel contexte), la production de mesures par la simulation, la mise en forme et la comparaison des résultats obtenus aux objectifs poursuivis. S'ils n'ont pas été atteints, de nouveaux scénarios sont proposés et testés jusqu'à satisfaction.

Dans la mesure où la plupart des modèles comportent des aléas, cette étape nécessite la détermination avec rigueur de la durée de simulation et du nombre de réplications (exécutions du modèle de simulation) ; elle fait appel aux outils statistiques afin de caractériser le comportement du modèle : calcul d'intervalles de confiance, de coefficients de corrélation, ...

1.2.6 Avantages et inconvénients de la simulation

- **Avantages**

1. Amélioration constante des processus : les processus sont visualisés dans un environnement virtuel afin de les appliquer ensuite dans le monde physique. La version numérique des processus étudiés est corrigée en permanence. Ce test continu, permet d'envisager de nouvelles opportunités. La version physique est enrichie de toute cette expérience.
2. Réduction des coûts : les créateurs et ingénieurs peuvent créer et perfectionner un produit dans un environnement virtuel, au lieu de créer des prototypes physiques qui entraîneraient un coût plus élevé.
3. Prévention des défaillances : Il est possible de détecter des problèmes avant qu'ils ne se produisent et d'éviter ainsi d'éventuelles interruptions du service et des temps d'arrêt durant la réparation des machines ou la modification des processus.
4. Planification de l'avenir : l'analyse de données et la simulation permettent de créer un comportement prédictif et, de promouvoir l'innovation dans tous les processus, accompagnant les entreprises à devancer leur concurrence.
5. Surveiller les décisions : tous les processus sont interconnectés, par conséquent, ils communiquent entre eux en temps réel, rendant possible la prise de décisions sur la base de données fiables et actualisées.
6. Augmentation de la productivité.

- **Inconvénients**

1. La conception de modèles peut nécessiter des compétences spéciales.
2. Résultats pas forcément généralisable.
3. Couteux

1.3 Génération des nombres aleatoires et pseudo-aléatoires

Pour la simulation d'un modèle non-déterministe, il est nécessaire de disposer d'une source de nombres susceptibles de jouer le rôle des variables aléatoires qui interviennent dans la définition du modèle.

1.3.1 Génération des nombres aléatoires

Un générateur de nombres aléatoires, random number generator (RNG) en anglais, est un dispositif capable de produire une séquence de nombres pour lesquels il n'existe aucun lien déterministe (connu) entre un nombre et ses prédécesseurs, de façon que cette séquence puisse être appelée « suite de nombres aléatoires ». Des méthodes pour obtenir des nombres aléatoires existent depuis très longtemps et sont utilisées dans les jeux de hasard : dés, roulette, tirage au sort, mélange des cartes, etc. Elles peuvent toutefois souffrir (et souffrent généralement) de biais.
Ces générateurs ont une utilité dans de nombreux domaines. Outre les jeux, on peut citer :

— La simulation.
— L'analyse.
— L'échantillonnage.
— La prise de décision.

1.3.2 Génération des nombres pseudo-aléatoires

Un générateur algorithmiques ou de nombres pseudo-aléatoires (GNPA) est un algorithme qui génère une séquence de nombres présentant certaines propriétés du hasard. De sorte que, les nombres générés par cet algorithme sont supposés être suffisamment indépendants les uns des autres, et qu'il est potentiellement difficile de repérer des groupes de nombres qui suivent une certaine règle.

Cependant, les sorties d'un tel générateur ne sont pas entièrement aléatoires ; elles s'approchent seulement des propriétés idéales des sources complètement aléatoires.

La raison pour laquelle on se contente de nombres pseudo-aléatoires est que :

— Il est difficile d'obtenir un grand échantillon de « vrais » nombres aléatoires.
— Les algorithmes générateurs sont particulièrement adaptés à une implémentation informatique, donc plus facilement et plus efficacement utilisables.

1.3.3 Génération des nombres uniformes

Pour aborder la simulation de la meilleur façon possible, on commence tout d'abord par introduire les générateurs de nombres uniformes (simulation de la loi uniforme sur l'intervalle [0,1], sur laquelle la simulation de toutes les autres lois est basée. La plupart des algorithmes générateurs de nombres pseudo-aléatoires ont pour but de produire des suites uniformément distribuées. Une classe très répandue de générateurs est basée sur la méthode des congruences.

1.3.3.1 La méthode des congruences

La méthode des congruences repose sur un algorithme simple pour la génération de nombres pseudo-aléatoires. En effet, cette derniére est basée sur la relation de récurrence suivante :

$$X_i = (aX_{i-1} + b) mod\ m \quad i = 1, 2, ..., n - 1.$$

où a, b, m et X sont des entiers positifs donnés. On dit que a est le multiplicateur, b est l'incrément, m le modulo, n le nombre des échantillons à générer et X_0 la valeur initiale appellée aussi graine. Le nombre pseudo-aléatoire, $U_i, i = 0, .., n - 1$, compris entre 0 et 1, est obtenu par la relation :

$$U_i = \frac{X_i}{m}.$$

La longueur d'une suite quelconque générée par une relation récursive du type $(aX_{i-1} + b)modm$ ne peut pas dépasser m puisqu'il y a au plus m nombres différents modulo m. De maniére générale, il faut choisir une grande valeur de m.

1.3.4 Quelques généarteurs informatiques

La plupart des générateurs de nombres aléatoires utilisés par les logiciels et les langages de programmation informatique sont construits à l'aide de la méthode de congruence présentée dans la section précédente. Le tableau(1.1), présente les paramètres de quelques générateurs utilisés en informatique.

Longage/Logiciel	Générateur	a	b	M
IBM	RANDU	$2^{16} + 3$	0	2^{31}
Scilab	RAND	843314861	453816693	2^{31}
Turbo Pascal	RANDOM	129	907633385	2^{32}
Langage c	RAND	103515245	12345	2^{31}
Maple	RAND	427419669081	0	$10^{12} - 11$

TABLE 1.1 – Générateurs informatique

Exemple 1.2. *L'algorithme ci-dessus, génère une suite de nombres aléatoires avec le générateur de Scilab.*

Algorithme
Debut
lire(n) ;
$a = 843314861$;
$b = 453816693$;
$m = 2^{31}$;
$X(1) = 0$; *%la graine*
pour $i = 1 : n$ ***faire***
$X(i+1) = (aX(i) + b) mod\ m$;
$U(i) = \frac{X(i+1)}{m}$;
fin pour ;
afficher $(U(i))$;
Fin.

La figure(1.3), montre la simulation de 1000 *tirages de variable aléatoire de loi* $\mathcal{U}([0,1])$ *obtenues par le générateur Scilab.*

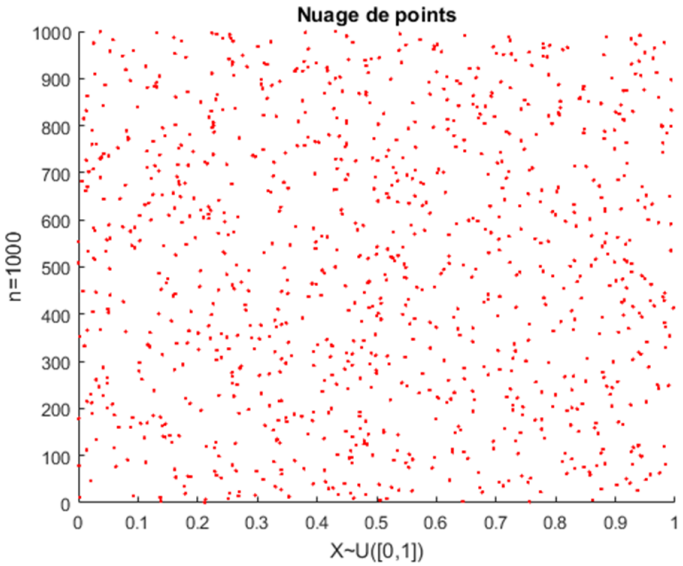


FIGURE 1.3 – Génération des nombres aléatoires.

Remarque 1.1. *Il existe plusieurs tests permettant de s'assurer de la qualité des nombres aléatoires produits par un générateur algorithmique. Ces derniers permettent de vérifier la stochasticité et l'uniformité des séquences* $(u_1, .., u_n)$ *générées. Parmis ces tests, on peut citer, le test de khi-deux et le test de Kolmogorov-Smirnov.*

1.4 Simulation de variables aléatoires

Dans la section précédente, nous avons vu comment simuler ou générer des suites de nombres aléatoires distribués de façon uniforme sur l'intervalle $[0, 1]$. Ceux-ci sont à la base de toute simulation. Dans cette section, nous présentons quelques méthodes

de génération de nombres non uniformes. Dans ce cas on parle de la simutaion d'une variable aléatoire qui suit une loi de probabilité quelconque.

1.4.1 Simulation de variables aléatoires discrètes

On présente ici quelques méthodes de simulation de variables aléatoires de lois discrètes.

1.4.1.1 Loi de Bernoulli de paramètre p notée $\mathcal{B}(p)$

La loi de Bernoulli est une distribution de probabilité discrète, qui prend la valeur 1 avec la probabilité p et 0 avec la probabilité $q = 1 - p$. En d'autres termes :

$$P(X = x) = \begin{cases} p & si\ x = 1 \\ 1 - p & si\ x = 0 \quad p \in [0, 1] \\ 0 & sinon \end{cases}$$

ou encore

$$P(X = x) = p^x (1 - p)^{1-x} \delta(x).$$

La simulation de la loi de Bernoulli est basée sur la proposition suivante :

Proposition 1.4.1. *Soit X une v.a.d suit la loi $\mathcal{B}(p)$, $X \sim \mathcal{B}(p)$, alors l'évènement $X \sim \mathcal{B}(p)$, peut s'écrire comme la fonction indicatrice d'un évènement $U \sim \mathcal{U}([0; 1])$ avec $U \leq p$. Autrement dit :*

$$X \sim \mathcal{B}(p) \Leftrightarrow X = \delta_{U \leq p}$$

preuve. On montre que $X = \delta_{U \leq p} \Rightarrow X \sim \mathcal{B}(p)$
Soit l'évènement $X = \delta_{U \leq p} \Leftrightarrow X = \begin{cases} 1 & si\ U \leq p \\ 0 & sinon \end{cases}$

— Si $X = 1$ alors $P(x = 1) = P(U \leq p) = \sum_{k=1}^{p} 1 = p$

— Si $X = 0$ alors $P(x = 0) = P(U > p) = 1 - P(U \leq p) = 1 - p$

d'ou la preuve de $X = \delta_{U \leq p} \Rightarrow X \sim \mathcal{B}(p)$.
Par réciproque, on en déduit que $X \sim \mathcal{B}(p) \Rightarrow X = \delta_{U \leq p}$

1.4.1.2 Loi binomiale (n, p) notée $\boldsymbol{Bin}(n, p)$

La loi binomiale, de paramètre n et p est la loi de probabilité d'une variable aléatoire X dont la fonction de masse est :

$$P(X = k) = C_n^k p^k (1 - p)^{n-k}, k = \{1, 2, ..., n\}.$$

La simulation d'une variable aléatoire qui suit une loi binomiale ($X \sim Bin(p)$) est basé sur la proposition suivante :

Proposition 1.4.2. *Si $X_1,\ X_2, ...,\ X_n$ sont n variables i.i.d suivant la loi Bernoulli $\mathcal{B}(p)$), la variable aléatoire $Y = X_1 + X_2 + ... + X_n$ suit une loi binomiale $Bin(n, p)$.*

preuve. Il est simple de montrer que :

$$\{X_1,\ X_2, ..., X_n\} i.i.d,\ X_i \sim \mathcal{B}(p) \Rightarrow Y = \sum_{i=1}^{i=n} X_i \sim Bin(n, p)$$

Il suffit donc de simuler n variables aléatoires indépendantes de loi $\mathcal{B}(p)$ et d'en faire leurs somme.

1.4.1.3 Loi de probabilité de Poisson

Afin de simuler une variable aléatoir X qui suit une loi de probabilité de Poisson, il existe deux méthodes :

Méthode 1

On sait que :

$$X \sim \mathcal{P}(\lambda) \Leftrightarrow p_k = F(X = k) = e^{-\lambda}\frac{\lambda^k}{k!}\ (Pour\ k \in \mathbb{N}),$$

ce qui implique que :

$$p_{k+1} = e^{-\lambda}\frac{\lambda^{k+1}}{k+1!} = \frac{\lambda}{k+1}p_k,$$

on note par P_k le cumul des p_k ;

$$P_k = \sum_{i=0}^{k} p_i,$$

donc

$$P_{k+1} = \sum_{i=0}^{k+1} p_i = P_k + p_{k+1} = P_k + \frac{\lambda}{k+1}p_k.$$

On simule donc une variable de Poisson de paramètre λ en prenant :

$$X = \sum_{k\geq 0} k.\delta_{P_{k-1}\leq U<P_k},$$

avec $P_{-1} = 0$.

Méthode 2

La deuxième méthode consiste à simuler des variables aléatoires $\{Y_1;\ Y_2; ...\}$ i.i.d. avec $Y_i \sim \mathcal{E}(\lambda)$ (voir le paragraphe sur les variables aléatoires continues) de sorte que X est définie par :

$$X = \sum_{k\geq 0} k.\delta_{Z_k\leq 1<Z_{k+1}},$$

avec

$$Z_k = \sum_{i=1}^{k} Y_i.$$

Donc il faut simuler des variables aléatoires exponentielles de paramètre λ et compter le nombre de simulations nécessaires pour dépasser 1, ou bien simuler des variables aléatoires exponentielles de paramètre 1 et compter le nombre de simulations nécessaires pour dépasser λ.

1.4.2 Simulation de variables aléatoires continues

Rappelons a juste titre, qu'il existe plusieurs méthodes pour la simulation de variables aléatoires continues. Dans cette section, nous nous contentons par la présentation des deux méthodes principales à savoir :

- La méthode d'inversion.
- La méthode d'acceptation rejet.

1.4.2.1 Méthodes d'inversion

Une des méthodes de simulation des variables aléatoires est la méthode d'inversion. Comme son nom l'indique cette méthode est fondée sur l'inversion de la fonction de répartition. Cette méthode est basée sur les deux théorème suivants :

Théorème 1. *Soit X une v.a de fonction de répartition F, continue et strictement croissante, on a :*
Si $U \sim \mathcal{U}([0,1])$ Alors $F^{-1}(U)$ à même loi que X.

Autrement dit, il suffit de simuler U suivant $\mathcal{U}([0,1])$ puis appliquer la transformation $X = F^{-1}(U)$ pour simuler suivant la loi de X.

preuve. Supposons que X continue,F croissante et F^{-1} existe, on pose : $X = F^{-1}(U)$

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x).$$

L'algorithme de cette méthode est donné comme suit :

Algorithme de simulation par inversion

Debut
lire(n) ; % taille d'échantillon
pour $i = 1 : n$ **faire**
  générer $U(i) \sim \mathcal{U}([0,1])$;
  $X(i) = F_X^{-1}(U(i))$;
fin pour ;
afficher $(X(i))$;
Fin.

Exemple 1.3. *Soit $X \sim \varepsilon(\lambda)$ une variable aléatoire de loi exponentielle de paramètre $\lambda > 0$.*
La densité de la loi exponentielle s'écrit

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0.$$

Et sa fonction de répartition est donée par :

$$F(x) = 1 - e^{-\lambda x}, \quad x > 0.$$

Posons pour tout $0 \leq u \leq 1$, $u = F(x)$ alors

$$u = 1 - e^{-\lambda x} \Leftrightarrow x = -\frac{\ln(1-u)}{\lambda}. \tag{1.1}$$

Si $U \sim \mathcal{U}([0,1])$, nous avons, d'aprés le résultat ci-dessus :

$$-\frac{\ln(1-U)}{\lambda} \sim \varepsilon(\lambda). \tag{1.2}$$

Remarquons que $(1-U)$ à même loi que U et donc

$$-\frac{\ln(U)}{\lambda} \sim \varepsilon(\lambda). \tag{1.3}$$

L'algorithme suivant permet de simuler un échantillon de variable aléatoire qui suit une loi exponentielle :

Algorithme de simulation de la loi $\varepsilon(\lambda)$

Debut
lire(n) ; % taille d'échantillon
lire(λ) ;
$i = 1$;
pour $i = 1 : n$ **faire**
  générer $U(i) \sim \mathcal{U}([0,1])$;
  $X(i) = -\frac{1}{\lambda}\log(U(i))$;
fin pour ;
afficher $(X(i))$;
Fin.

La figure (1.4), représente l'histogramme et la distribution d'un échantillon de taille $n = 1000$ généré à partir de la loi exponentielle de paramètre $\lambda = \frac{1}{2}$.

Remarque 1.2. *Nous avons vu précédemment que la méthode d'inversion requiert la connaissance et le calcul rapide de la fonction inverse de la fonction de répartition. Ceci n'est pas toujours envisageable. En effet, il suffit de penser à la loi normale centrée réduite de densité,*

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad x \in \mathbb{R},$$

dont la fonction de répartition n'admet pas d'expression analytique simple et encore moins pour sa fonction inverse.
Dans ce cas la méthode de rejet détaillée dans la sous-section suivante peut représenter une alternative judicieuse.

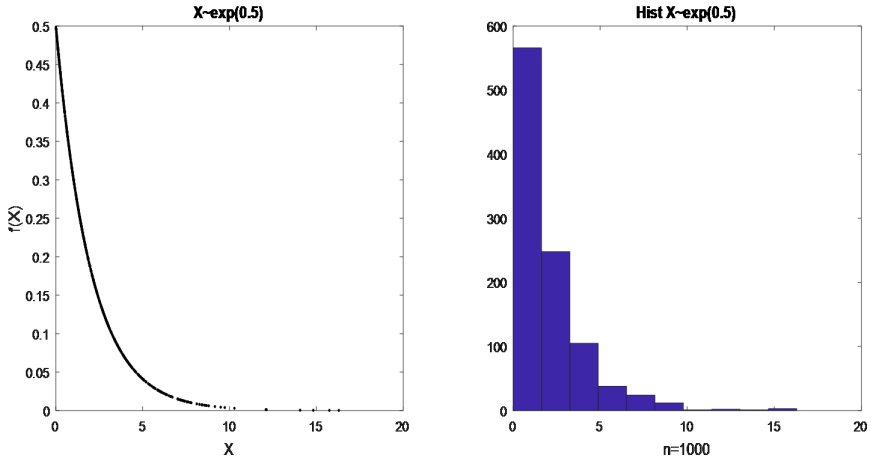


FIGURE 1.4 – Simulation de la loi exponentielle

1.4.2.2 Méthode d'acceptation-rejet

Il y en a plusieurs versions de la méthode de rejet. Nous nous contentons par présenter la méthode la plus générale. Supposons que l'on veuille simuler une variable aléatoire X de densité f (dans $\mathbb{R}^d$)et supposons aussi qu'il existe une loi de densité g facile à simuler telle que :

1. On sait simuler Y de densité g.
2. Il existe une constante m telle que, pour presque tout x ,$f(x) \leq mg(x)$.

Considérons alors deux suites indépendantes de variables aléatoires :

(a) $(Y_n)_{n\geq 1}$ iid de densité g.

(b) $(U_n)_{n\geq 1}$ iid de loi uniforme.

Autrement dit, Y correspond à une proposition et U à un tirage pile ou face pour décider si on accepte ou non cette proposition. Nous noterons par r la fonction rapport d'acceptation-rejet pour la pile ou face, à savoir pour tout $y \in \mathbb{R}^d$:

$$r(y) = \begin{cases} \frac{f(y)}{mg(y)}, & \text{si} \quad g(y) > 0; \\ 0, & \text{sinon.} \end{cases} \tag{1.4}$$

Par ailleurs, f et g étant des densités, la constante m intervenant dans la majoration est supérieure à 1.
La proposition suivante montre comment simuler suivant la densité f voulue.

Proposition 1.4.3. *Soit* $N = \inf\{n \geq 1, U_n \leq r(Y_n)\}$ *le premier instant où le tirage est accepté et* $X = Y_n$ *la valeur de la proposition à cet instant .Alors* X *a pour densité* f*. Par ailleurs,* N *suit une loi géométrique de paramètre* $\frac{1}{m}$*.*

Remarque 1.3. *La variable* N *étant géométrique de paramètre* $\frac{1}{m}$ *, elle a pour moyenne* m*. Concrètement, il faut donc faire en moyenne* m *essais pour obtenir une seule simulation selon la loi cible* f*. Dés lors, il s'agira de choisir le couple* (g, m) *de sorte que* m *soit aussi proche de* 1 *que possible.*
En d'autres termes. On a tout intérêt à choisir une densité g *qui ressemble le plus possible à* f*.*

L'algorithme de la méthode de rejet est donné comme suit :

Algorithme d'acceptation rejet

Debut
lire(n) ;
$i = 1$;
$j = 1$;
Tantque$(i \leq n)$**faire**
  générer $U(i) \sim \mathcal{U}([0, 1])$;
  générer $Y(i)$ selon g ;
  $r(i) = \frac{f(Y(i))}{mg(Y(i))}$;
  si $U(i) \leq r(i)$
  $X(j) = Y(i)$;
    $j = j + 1$;
  fin si ;
    $i = i + 1$;
fin Tantque ;
afficher (X) ;
Fin.

Afin de mieux comprendre la méthode d'acceptation-rejet on donne un exemple d'application complet.
Dans cet exemple, nous montrons comment déterminer la constante m et comment minimiser la probabilité de rejet (le nombre d'itération).

Exemple 1.4. *Simulation d'une loi normale centrée réduite à partir de la densité de Laplace.*
Soit f *la densité de la loi normale* $\mathcal{N}(0, 1)$*,*

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

$x \in \mathbb{R}$*.*

Soit g *la densité de la loi de Laplace,*

$$g(x) = \frac{\lambda}{2} e^{-\lambda|x|}.$$

Calculons le rapport $f(x)/g(x)$;
nous avons :

$$\forall x \in \mathbb{R},\ \frac{f(x)}{g(x)} = \frac{2}{\sqrt{2\pi}}\frac{1}{\lambda}e^{-\frac{x^2}{2}+\lambda|x|}$$

Par symétrie, nous pouvons écrire :

$$\forall x \geq 0,\ \frac{f(x)}{g(x)} = \frac{2}{\sqrt{2\pi}}\frac{1}{\lambda}e^{-\frac{x^2}{2}+\lambda x}.$$

Posons $h(x) = e^{-\frac{x^2}{2}+\lambda x}$ et étudions cette fonction. Pour tout $x \geq 0$, nous avons

$$h^{'}(x) = (-x+\lambda)h(x)$$

d'ou le tableau de variation de la table (1.2).

x	0	λ	$+\infty$
$h^{'}(x)$	$+$	0	$-$
$h(x)$	$\nearrow$	$h(\lambda)$	$\searrow$

TABLE 1.2 – Tableau de variation de h.

Donc, pour tout $x \in \mathbb{R}^+$,

$$f(x)/g(x) \leq \frac{2}{\sqrt{2\pi}}\frac{1}{\lambda}h(\lambda).$$

Posons :

$$m(\lambda) = \frac{2}{\sqrt{2\pi}}\frac{1}{\lambda}e^{\frac{\lambda^2}{2}}.$$

Il reste à minimiser la fonction $m(\lambda)$.
Nous avons $\forall \lambda \geq 0$,

$$m^{'}(\lambda) = \frac{2}{\sqrt{2\pi}}(1-\frac{1}{\lambda^2})e^{\frac{\lambda^2}{2}}.$$

D'où le tableau de variation de $m(\lambda)$ donné dans la table (1.3). D'après ce tableau, la valeur de λ qui permet de minimiser la probabilité de rejet est : $\lambda = 1$. Donc pour

λ	0	1	$+\infty$
$m^{'}(\lambda)$	$-$	0	$+$
$m(\lambda)$	$\searrow$	$m(1)$	$\nearrow$

TABLE 1.3 – Tableau de variation de m.

$\lambda = 1$, on touve $m = \frac{2}{\sqrt{2\pi}}$ et $g(x) = e^{-|x|}$.

D'après les résultats précédentes, on peut donc simuler f avec un algorithme de rejet (puisqu'il est facile de simuler suivant la loi de densité g.

1. ***Simulation de la loi de densité g :***
   Par l'application de la méthode d'inversion sur la fonction de répartition de g, on trouve :

$$\forall u \in [0,1], |y| = -log(u)$$

2. ***Le rapport d'acceptation-rejet*** $r(y)$
   Le rapport d'acceptation-rejet est donné par :

$$r(y) = e^{-\frac{(|y|-1)^2}{2}}.$$

La simulation d'une v.a qui suit la loi normale par la méthode de rejet est donnée par l'algorithme suivant :

Algorithme
Debut
lire(n) ;
$i = 1$;
$j = 1$;
Tantque$(i \leq n)$***faire***
  générer $U(i) \sim \mathcal{U}([0,1])$;
  $Y(i) = \log(2U(i))$;
  $r(i) = e^{-\frac{1}{2}(Y(i)-1)^2}$;
  si$(U(i) \leq r(i))$
    $X(j) = Y(i)$;
    $j = j + 1$;
  fin si ;
  $i = i + 1$;
fin Tantque ;
afficher $([X, -X])$;
Fin.

La figure (1.5), représente la simulation de la loi normale à partir de la loi de Laplace.

1.5 Conclusion

Dans ce chapitre nous avons introduit les concepts de base et les caractéristiques principales de la simulation stochastique dans le but de les utiliser pour l'évaluation des performances d'une file d'attente M/M/1.

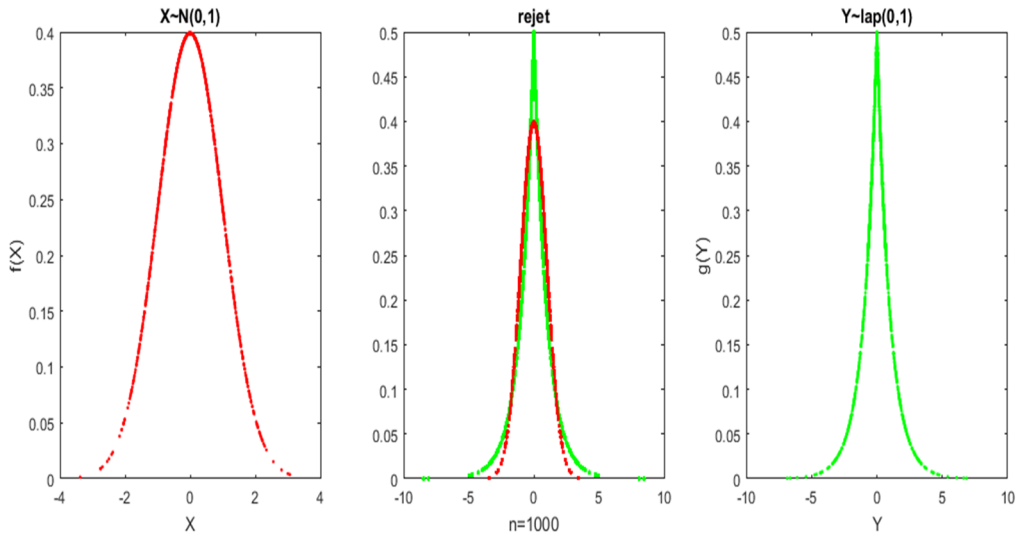


FIGURE 1.5 – Simulation d'une loi normale à partir de la loi de Laplace. A gauche, la représentation graphique de la densité de la loi normale, au milieu la simulation de la loi normale par la méthode de rejet et à droite la simulation de la loi de Laplace

Chapitre 2

Les méthodes d'échantillonnages

2.1 Introduction

Il est souvent difficile de résoudre analytiquement la plupart des problemes mathématiques rencontrés dans la vie de tous les jours, que ce soit en ingénierie, en economie, en sociologie ou en théorie de la décision. Or des résultats quantitatifs de plus en plus précis sont exigés par nos sociétés post industielles, ceci nous a conduits à introduire des modèles et des méthodes numériques pour répondre aux impératifs de la science appliquée, de l'industrie et de la societé. Ces méthodes, n'ont pris d'essor considerable ni été appliquées à une multitude de problémes pratiques que durant les dernieres années, avec l'amelioration des performances des ordinateurs, il est devenu plus efficace de simuler numériquement le comportement d'un système complexe que de l'observer expérimentalement. Pour ce faire, il existe plusieurs méthodes numériques. Dans notre travail l'interet sera porté sur les méthodes stochatiques.

Dans ce chapitre, notre présentation est axée sur les differentes méthodes d'echantillonnages à savoir l'échantillionnage aléatoire (méthode de Monte Carlo), l'échantillonnage descriptif et l'échantillonnage descriptif amélioré.

2.2 Méthodes de Monte Carlo

Les méthodes dites de Monte Carlo, dont le principe repose essentiellement sur les lois des grands nombres, c'est-à-dire sur une répétition d'expériences aléatoires, permettent d'évaluer une quantité donnée mais inconnue, voir de résoudre un système déterministe. Ces méthodes peuvent servir pour :

— le calcul d'intégrale,
— la résolution d'équations aux dérivées partielles,
— la résolution de systèmes linéaires,

— la résolution de problémes d'optimisation.
Dans ce qui suit,nous nous intéressons à une application classique des méthodes de Monte-Carlo à savoir l'intégration de Monte-Carlo. Cette dernière consiste à utiliser des méthodes de simulation pour estimer d'une manière approchée les quantités du type :

$$I = E[\varphi(X)] = \int_{\mathbb{R}^d} \varphi(x) f(x) dx \quad d \in \mathbb{N}^*. \tag{2.1}$$

Où $\varphi : \mathbb{R}^d \longrightarrow \mathbb{R}$ est une fonction donnée et X un vecteur aléatoire de densité f suivant laquelle on sait simuler.
Dans ce contexte, l'estimateur Monte-Carlo de base est défini par :

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} \varphi(X_i). \tag{2.2}$$

Où les X_i sont générées de façon i.i.d. selon f.

C'est la loi forte des grands nombres qui permet de justifier la convergence de la méthode.

Théorème 2. *(Convergence)*
Soit $(X_n)_{n\geq 1}$ une suite de v.a.i.i.d à valeur dans $\mathbb{R}^d$, $d \in \mathbb{N}^*$, $X_i \sim \mathcal{L}(f)$. On suppose que $\varphi(x)$ est une fonction mesurable, $(E[\varphi(X)] < +\infty)$, alors :

$$\frac{1}{n} \sum_{i=1}^{n} \varphi(X_i) \underset{n\to+\infty}{\overset{p.s}{\longrightarrow}} E[\varphi(X)].$$

Ce théorème montre pourquoi l'approximation de Monte-Carlo (2.2) est valide (et sous quelle hypothèse).

Autrement dit, afin d'estimer une quantité de type $I = \int_{\mathbb{R}^d} \varphi(x) f(x) dx$ avec une méthode de Monte-Carlo on doit suivre les étapes suivantes :

Etape1 : Si φ est une fonction mesurable $(E[\varphi(X)] < +\infty)$, alors on peut mettre la quantité I, sous forme d'une espérance, i.e.

$$I = \int_{\mathbb{R}^d} \varphi(x) f(x) dx = E[\varphi(X)].$$

Etape2 : Si on sait simuler $\varphi(X_1), ..., \varphi(X_n)$, alors par l'application de la loi forte des grands nombres nous pouvons approcher la quantité I par un estimateur Monte-Carlo, i.e.

$$I = E[\varphi(X)] \approx \frac{1}{n} \sum_{i=1}^{n} \varphi(X_i).$$

Sous les hypotèses données précédemment et les lois des grands nombres, l'algorithme qui permet d'estimer une quantité de type $I = \int\limits_{\mathbb{R}^d} \varphi(x) f(x) dx$ est donné comme suit :

Algorithme

Debut
lire(n) ;
$s = 0$;
pour $i = 1 : n$ **faire**
générer $X(i) \sim \mathcal{L}(f)$;
$s = s + \varphi(X(i))$;
fin pour ;
$\hat{I}_n \approx \frac{s}{n}$;
afficher $(\hat{I}_n)$;
Fin.

2.2.1 Exemples d'application

Exemple 2.1. ***Estimation de la valeur de*** π
Soit une suite de couple $(X_n, Y_n)_{n>0}$ *de v.a.i.i.d* $(X_1, Y_1) \sim \mathcal{U}([-1, 1])$.
Posons $\varphi(x, y) = 1_{x^2+y^2\leq 1}$.
L'aire du cercle d'unité est donnée par :

$$\begin{aligned} I &= \int_{-1}^{1}\int_{-1}^{1} 1_{x^2+y^2\leq 1} dxdy \\ &= 4\int_{-1}^{1}\int_{-1}^{1} 1_{x^2+y^2\leq 1} dxdy \times \frac{1}{2}1_{-1\leq x\leq 1} \times \frac{1}{2}1_{-1\leq y\leq 1} dxdy \\ &= 4\int_{-1}^{1}\int_{-1}^{1} \varphi(x, y) f(x, y) dxdy \end{aligned}$$

D'après la loi des grands nombres on aura,

$$\frac{1}{n}\sum_{i=1}^{n} \varphi(X_i, Y_i) \xrightarrow[n\to+\infty]{p.s} E[\varphi(X, Y)].$$

D'où $I \approx 4 \times \frac{1}{n}\sum_{i=1}^{n} \varphi(X_i, Y_i)$. *Mais, nous savons que l'air du cercle d'unité est* π, *d'où*

$$\pi \approx 4 \times \frac{1}{n}\sum_{i=1}^{n} \varphi(X_i, Y_i).$$

Un des algorithmes d'estimation de la valeur de π *est donné comme suit :*

Algorithme
Debut
lire (n) ;
$s = 0$;
pour $i = 1 : n$ ***faire***
générer $U \sim \mathcal{U}([0,1])$;
générer $V \sim \mathcal{U}([0,1])$;
$X(i) = 2 * U - 1$;
$Y(i) = 2 * V - 1$;
si $(X(i)^2 + Y(i)^2 \leq 1)$;
$s = s + 1$;
fin si ;
fin pour ;
$\pi = 4 * \frac{s}{n}$;
Fin.

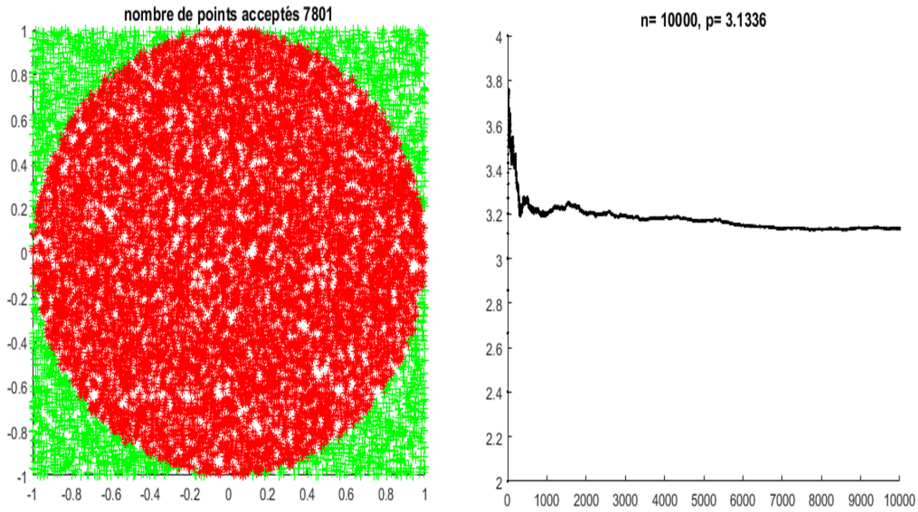


FIGURE 2.1 – Estimation de la valeur de π.

2.3 Erreur de l'estimation M.C

L'estimation d'une quantité $E[\varphi(X)]$ par une méthode de Monte-Carlo génère une erreur aléatoire dont on ne peut pas la borner. En revanche, on peut donner un intervalle de confiance au résultat grâce au théorème centrale limite (TCL).

Théorème 3. *TCL*
Soit $\varphi(X_i),\ i = 1...n$, *une suite de v.a.i.i.d à valeur dans* $\mathbb{R}^d,\ d \in \mathbb{N}^*$.

On suppose que $E[\varphi(X)^2] \leq +\infty$.

Soit $var[\varphi(X)] = \sigma^2$ *(Variance de* $\varphi(X)$*), alors*

$$\frac{\sqrt{n}}{\sigma}\left[\frac{1}{n}\sum_{i=1}^{n}\varphi(X_i) - E[\varphi(X)]\right] \xrightarrow[n\to+\infty]{loi} \mathcal{N}(0,1).$$

Cela signifier que la vitesse de convergence de la méthode de monte-carlo est de l'ordre de $\frac{\sigma}{\sqrt{n}}$.

Notons l'erreur d'estimation $\varepsilon_n = |\hat{I}_n - I| = |\frac{1}{n}\sum_{i=1}^{n}\varphi(X_i) - E[\varphi(X)]|$.

Sous les hypothèses du théorème limite centrale et pour $\alpha \in [0,1]$ fixé. Un intervalle de confiance de niveau asymptotique $1-\alpha$ pour I est donné par :

$$\left[\hat{I}_n - \phi^{-1}(1-\alpha/2)\sqrt{\frac{\sigma^2}{n}},\ \hat{I}_n + \phi^{-1}(1-\alpha/2)\sqrt{\frac{\sigma^2}{n}}\right].$$

Où $\phi^{-1}(1-\alpha/2)$ désigne le quantile d'ordre $1-\alpha/2$ de la loi normale centrée réduite.

2.3.1 Les erreurs générés par la méthode de Monte-Carlo

Généralement, il existe deux types des erreurs générés par l'application d'une méthode de monte-carlo à savoir : les erreurs ensembliste et les erreurs seuentiels.

Erreur ensembliste

Comme son nom l'indique, l'erreur ensembliste est liée a l'ensemble des valeurs produites par la loi uniforme. Cette erreur est caractérisée par la difference entre la distribution empirique et la distribution théorique ou la difference entre le graphe de la série de données (histogramme) et le graphe de la fonction de probabilité. L'effet ensembliste est controlé si la procedure d'échantillonnage est parfaitement maitrisée.

Erreur sequentiel

L'erreur séquentiel, est induite par la suite de l'ensemble des valeurs produite par la loi uniforme. Cette erreur est caractérisée par l'ordre dans lequel les valeurs de la loi uniforme se présentent. Contrairement a l'effet ensembliste, l'effet seguentiel est difficile à contrôler en simulation à l'exception de quelque cas.

Après avoir introduit les erreurs de l'estimation Monte-carlo, nous nous interessons maintenant à quelques méthodes d'echantillonnages qui permettent d'améliorer l'erreur de précision.

2.4 Méthodes de réduction de variance

Nous avons vu que la vitesse de convergence de la méthode de Monte Carlo est de l'ordre de $\frac{\sigma}{\sqrt{n}}$. Pour améliorer cette méthode il existe de nombreuses techniques, dites

réduction de variance, qui cherchent à diminuer la valeur de σ^2. Nous allons passer en revue plusieurs de ces méthodes [6].

2.4.1 Méthode d'échantillonnage préférentiel

Supposons qu'on cherche à estimer la valeur de $E[\varphi(X)]$ où X est une variable à valeur dans $\mathbb{R}^d$, de densité f.

Pour toute densité $h \geq 0$ nous pouvons écrire

$$\begin{aligned} E[\varphi(X)] &= \int_{\mathbb{R}^d} \varphi(x) f(x) dx \\ &= \int_{\mathbb{R}^d} \frac{\varphi(x) f(x)}{h(x)} h(x) dx \\ &= E\left[\frac{\varphi(Y) f(Y)}{h(Y)}\right]. \end{aligned}$$

où Y est une variable à valeur dans $\mathbb{R}^d$ de densité h.

Affin d'appliquer cette méthode, on doit choisir une densité h. Le choix le plus simple de h est $h : y \longmapsto \frac{\varphi(y) f(y)}{E[\varphi(Y)]}$. Ce qui bien une densité de probabilité.

Afin de comparer entre les deux méthodes, on doit comparer entre les variances de ces deux dernières. En effet, nous pouvons dire que la méthode d'échantillonnage préférentielle est plus intéressante que la méthode de Monte-Carlo classique si

$$var\left[\frac{\varphi(Y) f(Y)}{h(Y)}\right] \leq var[\varphi(X)].$$

Nous avons alors

$$var\left[\frac{\varphi(Y) f(Y)}{h(Y)}\right] = E\left[\left(\frac{\varphi(Y) f(Y)}{h(Y)}\right)^2\right] - E\left[\left(\frac{\varphi(Y) f(Y)}{h(Y)}\right)\right]^2. \tag{2.3}$$

Nous avons $h(y) = \frac{\varphi(y) f(y)}{E[\varphi(Y)]}$

Par substitution dans l'équation (2.3), on trouve :

$$var\left[\frac{\varphi(Y) f(Y)}{h(Y)}\right] = E[(\varphi(Y))]^2 - E[\varphi(Y)]^2 = 0.$$

Nous avons ici une méthode de Monte-Carlo de variance nulle, ce qui semble n'avoir aucun sens. Donc, il est de notre intérêt de choisir une autre densité h. L'idée donc serais de trouver une autre densité h.

La discussion ci-dessus nous donne une idée d'une démarche à suivre pour le choix de la densité h et ainsi réduire la variance.

1. Trouver une fonction h_1 grossièrement proche de $|\varphi.f|$ telle que l'on sache simuler suivant la densité :
$$h(y) = \frac{h_1(y)}{\int\limits_{\mathbb{R}^d} h_1(y) dy}.$$

2. On suppose que $Y_i,\ i = 1..n$ est une suite de v.a.i.i.d, $Y_1 \sim \mathcal{L}(h)$. Puis par l'application des lois des grands nombres,on pose

$$I \approx \hat{I}_n^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{\varphi(Y_i)f(Y_i)}{h(Y_i)}$$

2.4.2 Méthode de Variable de Contrôle

Nous cherchons à calculer $E[\varphi(X)]$ avec X variable à valeur dans $\mathbb{R}^d$ de densité f. Si on sait calculer (pas approcher) la quantité $E[h(X)]$, d'une certaine fonction h, alors on peut aussi faire l'estimation suivante :

$$E[\varphi(X)] \approx \hat{I}_n^2 = \frac{1}{n}\sum_{i=1}^{n}[\varphi(X_i) - h(X_I)] + E[h(X)].$$

En effet, nous avons

$$\begin{aligned} E[\varphi(X)] &= \int \varphi(x)f(x)dx \\ &= \int \varphi(x)f(x) - f(x)h(x) + f(x)h(x)dx \\ &= \int (\varphi(x) - h(x))f(x)dx + \int f(x)h(x)dx \\ &= \frac{1}{n}\sum_{i=1}^{n}[\varphi(X_i) - h(X_I)] + E[h(X)]. \end{aligned}$$

Donc la méthode de variable de contrôle est la suivante.

1. Trouver une fonction h proche de φ, telle que l'on sache calculer $E[h(X)]$ (le fait que h soit proche de φ signifié que la quantité $var[\varphi(X) - h(X)]$ est petite).
2. Estimer les variances $var[\varphi(X)]$ et $var[\varphi(X) - h(X)]$ et les comparer.

2.4.3 Méthode de variable antithétique

Supposons que l'on cherche à estimer la quantité suivante :

$$\begin{aligned} I &= \int_{[0,1]^d} \varphi(x_1, ..., x_d)dx_1...dx_d \\ &= \int_{[0,1]^d} \varphi(x_1, ..., x_d)f(x_1)...f(x_d)dx_1...dx_d. \end{aligned}$$

avec $f(x_d) = 1_{[0,1]}(x_d)$.
Pour $U \sim \mathcal{U}([0,1])$, nous avons $(1, ..., 1) - U$ est de même loi que U, alors nous pouvons écrire

$$I \approx E\left[\frac{\varphi[(1, ..., 1) - U] + \varphi(U)}{2}\right].$$

Ce qui donne l'idée pour une deuxième méthode.

$$I \approx \hat{I}_n^2 = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\varphi[(1,...,1)-X_i]+\varphi(X_i)}{2}\right].$$

Lemme 2.1. *Sous les hypothèses ci-dessus, nous avons,*

$$var\left[\frac{\varphi[(1,...,1)-U]+\varphi(U)}{2}\right] \leq var[\varphi(U)].$$

La variance est donc toujours réduite.

preuve. Nous avons,

$$var\left[\frac{\varphi[(1,...,1)-U]+\varphi(U)}{2}\right]$$

$$\begin{aligned}
&= E\left[\left(\frac{\varphi[(1,...,1)-U]+\varphi(U)}{2}\right)^2\right] - \underbrace{E\left[\frac{\varphi[(1,...,1)-U]+\varphi(U)}{2}\right]^2}_{A} \\
&= \frac{1}{4}\quad E\left[\varphi[(1,...,1)-U]^2+\varphi(U)^2+2\varphi[(1,...,1)-U]\varphi(U)\right]-A \\
&= \frac{1}{4}\quad E[\varphi[(1,...,1)-U]^2]+\frac{1}{4}\quad E[\varphi(U)^2]+\frac{1}{2}\quad E[\varphi[(1,...,1)-U]\varphi(U)]-A
\end{aligned}$$

On a $E(a\times b)\leq E(a^2)^{1/2}\times E(b^2)^{1/2}$ (Inégalité de Schwarz)

$$\begin{aligned}
var\left[\frac{\varphi[(1,...,1)-U]+\varphi(U)}{2}\right] &\leq \frac{1}{4}E[\varphi[(1,...,1)-U]^2]+\frac{1}{4}E[\varphi(U)^2] \\
&+ \frac{1}{2}\left[E[\varphi[(1,...,1)-U]^2]^{1/2}]\times E[\varphi(U)^2]^{1/2}\right]-A
\end{aligned}$$

nous avons $(1,...,1)-U\xrightarrow{loi}U$.

$$\begin{aligned}
&\leq \frac{1}{4}E[\varphi(U)^2]+\frac{1}{4}E[\varphi(U)^2]+\frac{1}{2}\left[E[\varphi(U)^2]^{1/2}]\times E[\varphi(U)^2]^{1/2}\right]-E[\varphi(U)]^2 \\
&\leq \frac{1}{2}E[\varphi(U)^2]+\frac{1}{2}E[\varphi(U)^2]-E[\varphi(U)]^2 \\
&\leq var[\varphi(U)].
\end{aligned}$$

Remarque 2.1. *Le même résultat est encore valable si on remplace l'intervalle* $[0,1]^d$ *par un domaine quelconque* E *de* $\mathbb{R}^d$, $d\in\mathbb{N}^*$, *et* $(1-u_1,...,1-u_d)\in[0,1]^d$ *par une transformation bijective* $g:E\to E$ *telle que pour tout* $x\in E$, $\varphi(g(x))=\varphi(x)$.

2.4.4 Méthode de stratification

La méthode de stratification est une méthode bien connue des statisticiens sondeurs. Supposons que l'on cherche à calculer
$I = E(\varphi(x)) = \int \varphi(x) f(x) dx$ où X suit la loi de f, on commence par décomposer I en :

$$I = \sum_{i=1}^{m} I_i = \sum_{i=1}^{m} E\left(1_{\{X \in D_j)} \varphi(x)\right)$$

Telle que D_i est une partition de l'ensemble sur lequel on intègre. On utilise alors n_i tirages pour estimer l'integrale I_i . Si l'on note $\delta_i^2 = \text{var}\left(1\left\{x \in D_i\right\} \varphi(x)\right)$ il est facile de voir que la variance de l'approximation est donnée par :

$$\sum_{i=1}^{m} \frac{\sigma_i^2}{n_i}.$$

Et si on minimise cette erreur à $\sum_{i=1}^{m} n_i = n$ fixé, on obtient $n_i = n \frac{\sigma_i}{\sum_{i=1}^{m} \sigma_i}$.
Le minimum vaut alors $\frac{1}{n}\left(\sum_{i=1}^{m} \sigma_i\right)$. On peut montrer qu'il est inferieur à la variance que l'on obtiendrait avec n tirages aléatoires de Monte-Carlo. Evidemment on ne peut que rarement calculer les σ_i, ce qui limite la portée de cette technique, mais on ne peut toujours les estimer à l'aide d'une méthode de Monte-Carlo.

2.4.5 Valeur moyenne

Supposons que l'on cherche à calculer :

$$E(\varphi(X, Y)) = \int \varphi(x, y) f(x, y) dx dy$$

Où $f(x, y)$ est la loi du couple (X, Y).

Si l'on pose $h(x) = \frac{1}{m(x)} \int \varphi(x, y) f(x, y) dy$ avec $m(x) = \int f(x, y) dy$, on aura :

$$E(\varphi(X, Y)) = E(h(X))$$

En effet la loi de X est m, et donc :

$$E(h(X)) = \int m(x) h(x) dx = \int [\int \varphi(x, y) f(x, y) dy] dx = E(\varphi(X, Y)).$$

On peut d'autre part montrer, en interprétant $h(x)$ comme une espérance conditionnelle, que :

$$\text{var}(h(X)) \leq \text{var}(\varphi(X, Y)),$$

Même si l'on peut calculer explicitement la fonction $h(x)$,il est préférable d'utiliser une méthode de Monte-carlo pour $h(x)$.

2.5 Méthode d'échantillonnage descriptif : ED (descriptive sampling : DS)

L'échantillonnage descriptif est une technique de Monte-Carlo, basée sur un choix déterministe des valeurs d'échantillon. Cette technique représente un changement conceptuel profond sur la façon dont nous effectuons une application de Monte Carlo. Elle s'inscrit dans la direction de l'abandon de paradigme qu'un choix aléatoire des valeurs d'un échantillon est nécessaire afin de décrire un comportement aléatoire.

Cette méthode a été proposée par SALIBY (1980. 1990) comme approche alternative a la méthode de Monte-Carlo. L'ED évite la variabilité des valeurs de l'échantillon et mènes à des évaluations plus précises dans une simulation, l'ED réduit aussi la variance. L'apport de cette approche est prouvé dans plusieurs travaux, SALIBY [5, 6, 7]. Malgré les avantages de cette méthode (elle évite l'effet ensembliste de Monte Carlo), l'effet séquentiel demeure toujours.

2.5.1 Procédure d'échantillonnage descriptif

Supposons que le problème de simulation contient une variable aléatoire d'entrée X de fonction de répartition F, et soit F^{-1} sa fonction inverse telle que $X = F^{-1}(R)$ où $R \sim \mathcal{U}([0,1])$. La génération d'un échantillon descriptif, se fait la manière suivant :

1. On définit $r_i = \frac{i-0.5}{n},\ i = 1, ..., n$. Cela permet de subdiviser l'intervalle $[0,1]$ en n sous intervalles équiprobables, tel que r_i représente le milieu de $i$ème sous intervalle.
2. Calculer les valeurs de l'échantillon x_i telles que :

$$x_i = F^{-1}(r_i) \text{ Pour } i = 1, \ldots, n.$$

3. Permuter aléatoirement l'ensemble des valeurs $x_i, i = 1, \ldots, n$.
4. Stocker la suite obtenue dans un nouveau vecteur Y et l'utiliser lorsque c'est nécessaire.

Exemple 2.2. *Estimation de la moyenne de la loi exponentielle de paramètre $\lambda = 1$, pour $n = 5$. Nous avons :*

$$r_i = \frac{i - 0.5}{n},\ i = 1, ..., 5.$$

Avec la méthode d'inversion (vue précédemment) on obtient :

$$x_i = -ln(1 - r_i),\ i = 1, ..., 5.$$

Par substitution, on obtient le tableau suivant :

i	r_i	x_i
1	0.1	0.1053
2	0.3	0.3566
3	0.5	0.6931
4	0.7	1.2039
5	0.9	2.3025
Total		4.6614

TABLE 2.1 – Résultats de l'exemple de la méthode ED

L'éstimateur de la moyenne est :

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{4.6614}{5} = 0.93228.$$

Néanmoins l'application de cet algorithme peut s'avérer trop couteuse a cause de la croissance de la taille de l'échantillon et les resultats de la simulation obtenue peuvent être biaisés.

2.5.2 Echantllion descriptif

Un échantillon descriptif de taille n est représenté par un ensemble de n variables aléatoires pris dans un ordre aléatoire $\{xd_1, \ldots, xd_n\}$.
Après permutation, ces variables sont appelées : variables descriptives. Un échantillon descriptif est un échantillon dont les observations ne sont pas des variables aleatoires. entièrement indépendantes mais chacune d'elle suit la distribution de la population.
La procédure d'échantillonnage descriptif nécessite la considération d'échantillons indépendants uniformément distribués sur l'intervalle $[0, 1]$. De ce fait, un générateur congruentiel est utilisé pour la permutation de l'ensemble des valeurs de la variable d'entrée.
Dans différentes histoires de simulation, la procédure de géneration d'échantillons descriptifs utilise le *même* ensemble des valeurs de la variable d'entrée mais pris dans un ordre aléatoire différent. Contrairement *à* l'échantillonnage aléatoire *où* cet ensemble varie d'une histoire à une autre.

Remarque 2.2. *D'après Saliby [5], les variables descriptives sont négativement corrélées et le coefficient de corrélation entre deux différentes variables descriptives est :*

$$\rho = -\frac{1}{n-1} = \mathrm{O}\left(\frac{1}{n}\right)$$

2.6 Méthode d'échantillonnage descriptif amélioré EDA

Cette méthode est basée sur le principe de la méthode définit précédemment (ED), elle élimine les inconvénients cités plus haut. cette méthode est définit par un bloc de sous-ensembles de nombres réguliers dont les tailles sont des nombres premiers. Ces derniers sont choisis aléatoirement. Ce bloc doit être situé à lintérieur d'un générateur distribuant ces nombres réguliers a la demande de la simulation. La procedure s'arrête

lorsque la simulation se termine.
Chaque histoire est déterminée par un bloc de nombres premiers differents. Si la simulation nécessite M histoires répliquées, nous considérons alors M blocs de $m_1, \ldots, m_M$ sous ensembles réguliers. Les nombres premiers et les valeurs des sous-ensembles d'entrée ne sont pas les mêmes pour toutes les histoires répliquées.

Remarque 2.3. *Par construction, cette méthode ne nécessite pas la connaissance à priori de la taille de l'échantillon.*

2.6.1 Histoire de simulation

Soient $p_q, q = 1, 2, 3, .., m$ des nombres premiers distincts.
Supposons que la variable de sortie possède k paramètres à estimer. Supposons aussi que la simulation se termine quand m nombres premiers sont utilisés et on obtient m sous histoires.

Dans une histoire donnée, la méthode mène aux m estimateurs des paramètres $\theta_j, j = 1, 2, \ldots, k$ suivants qui sont observés dans les m sous histoires :

$$\begin{aligned}
(Yr)_j^1 &= F_j\left(r_1^1, r_2^2, \ldots, r_{p_1}^{p_1}\right) \quad j = 1, 2, \ldots, k \\
(Yr)_j^2 &= F_j\left(r_{1+p_1}^1, r_{2+p_1}^2, \ldots, r_{p_1+p_2}^{p_2}\right) \quad j = 1, 2, \ldots, k \\
(Yr)_j^m &= F_j\left(r_{1+\sum_{i=1}^{m-1} p_i}^1, r_{2+\sum_{i=1}^{m-1} p_i}^2, \ldots, r_{\sum_{i=1}^{m} p_i}^{p_m}\right) \quad j = 1, 2, \ldots, k
\end{aligned}$$

Par convention

$$\sum_{i=0}^{m} p_i = 0.$$

Les sous ensembles de nombres réguliers

$$\left(r_{1+\sum_{i=1}^{q-1} p_i}^1 r_{2+\sum_{i=1}^{q-1} p_i}^2, \ldots, r_{\sum_{i=1}^{q} p_i}^{p_q}\right) \quad q = 1, 2, \ldots, m.$$

Sont dépendants, uniformément distribués sur $[0, 1]$ et dont les tailles sont des nombres premiers $p_1,\ p_2 \ldots, p_n$

Comme dans l'échantillonnage descriptif, ces nombres réguliers s'obtiennent de la manière suivante :

$$r_{i+\sum_{i=1}^{q-1} p_i}^i = \frac{i-0.5}{p_q} \quad i = 1, 2, \ldots, p_q \text{ et } q = 1, 2, \ldots, m$$

En conclusion, dans une histoire de simulation, la méthode d'échantillonnage descriptif amélioré génère les estimateurs de $\theta_j, j = 1, 2, \ldots, k$, de la maniere suivante :

$$(Yr)_j = \frac{1}{m} \sum_{i=1}^{n} (Yr)_j^i \quad j = 1, 2, \ldots, k$$

Ces derniers représentent la moyenne des estimateurs $(Y_r)_i^j, i = 1, 2, \ldots, m$.

2.6.2 La génération des valeurs des sous ensembles

Les valeurs de la variable aléatoire en entrée sont générées à la demande de la simulation.
La méthode d'inversion offre les valeurs régulières des sous ensembles qui sont données par :

$$(xd)^i_{i+\Sigma_{i=1}^{q-1} p_i} = F^{-1}\left(r^i_{i+\sum_{i=1}^{q-1} p_i}\right) \text{ pour } i = 1, 2, \ldots, p_q \text{ et } q = 1, 2, \ldots, m \qquad (2.4)$$

Exemple 2.3. *Estimation de la moyenne de la loi exponentielle de paramètre $\lambda = 1$, par la méthode d'EDA.*
Les sous ensembles de nombres réguliers sont donnés par la relation suivante :

$$\left(r^1_{1+\sum_{i=1}^{q-1} p_i} r^2_{2+\sum_{i=1}^{q-1} p_i}, \ldots, r^{p_q}_{\sum_{i=1}^{q} p_i}\right) \quad q = 1, 2, \ldots, m,$$

où $p_1,\ p_2, ..., p_q$ sont des nombres premiers.

A partir de l'équation(2.4) et par l'application de la méthode d'inversion on obtient :

$$(xd)^i_{i+\Sigma_{i=1}^{q-1} p_i} = -ln(1 - r^i_{i+\sum_{i=1}^{q-1} p_i}) \text{ pour } i = 1, 2, \ldots, p_q \text{ et } q = 1, 2, \ldots, m, \qquad (2.5)$$

on génère trois nombres premier, $p_1 = 7,\ p_2 = 11$ et $p_3 = 13$. Par substitution dans les deux équations (2.4) et (2.5), nous obtenons les résultats donnés dans le tableau (2.2).

i	r^i_i	$(xd)^i_i$	r^i_{i+7}	$(xd)^i_{i+7}$	r^i_{i+7+11}	$(xd)^i_{i+7+11}$
1	0.071	0.074	0.045	0.047	0.038	0.039
2	0.214	0.136	0.241	0.147	0.115	0.123
3	0.357	0.442	0.227	0.248	0.192	0.214
4	0.5	0.693	0.318	0.383	0.269	0.314
5	0.643	1.030	0.409	0.526	0.346	0.425
6	0.786	1.540	0.5	0.693	0.423	0.550
7	0.927	2.639	0.591	0.894	0.5	0.693
8		T_1=6.659	0.682	1.145	0.577	0.860
9			0.773	1.482	0.654	1.061
10			0.864	1.992	0.731	1.312
11			0.955	3.091	0.808	1.649
12				T_2=10.647	0.885	2.159
13					0.962	3.258
						T_3=12.662

TABLE 2.2 – Résultats de l'exemple de la méthode EDA

Nous calculons les trois moyennes $(Y_r)^i_j, i = 1, 2, 3.$

$$(Y_r)^1_1 = \frac{T_1}{p_1} = \frac{6.657}{7} = 0.951$$

$$(Y_r)^2_1 = \frac{T_2}{p_2} = \frac{10.648}{11} = 0.968$$

$$(Y_r)^3_1 = \frac{T_3}{p_3} = \frac{12.662}{13} = 0.974.$$

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

$$(Yr)_1 = \frac{1}{3}\sum_{i=1}^{3}(Yr)_1^i = 0.965.$$

2.6.3 Les sous-ensembles descriptifs

La procédure permettant d'obtenir les sous ensembles descriptifs dont les tailles sont des nombres premiers p_q est la suivante :

1. Générer les sous-ensembles de nombres réguliers

$$\left(r^1_{1+\sum_{i=1}^{q-1} p_i}, r^2_{2+\sum_{i=1}^{q-1} p_i}, \cdots, r^{p_i}_{p_i+\sum_{i=1}^{q-1} p_i}\right) \quad q = 1, 2, \ldots, m,$$

   dont les tailles sont des nombres premiers p_q.
2. Permuter les différentes séquences obtenues.
3. Calculer, à la demande de la simulation, les valeurs régulières des sous ensembles de variable aléatoire en entrée,

$$(xd)^i_{i+\sum_{i=1}^{q-1} p_i} \text{pour } i = 1, 2, \ldots, p_q.$$

Cette façon de génération est motivée par le fait que la simulation peut se terminer avant l'utilisation complète du dernier sous ensemble descriptif. Ceci diminue le côut de l'expérience car on ne calcule qu'une partie des valeurs du dernier sous ensemble descriptif généré.

2.6.4 La structure de données

Pour chaque variable aléatoire d'entrée, on définit un enregistrement avec la structure suivante :

— p : Un nombre premier représentant la taille du sous ensemble de nombres réguliers.
— R : Un vecteur réel $(1, \ldots, p)$, contenant le sous ensemble de nombres réguliers.
— ip : Un entier pointant vers le premier élément disponible r à générer. Si $ip = 1$, aucun élément n'a encore été généré. Si $ip > p$, tout le sous ensemble de nombres réguliers a déjà été généré .

2.6.5 Algorithme d'EDA

(a) Initialisation de l'experience :

1. Générer un nombre premier aléatoirement.
2. Génerer le sous ensemble de nombres réguliers $r_i, i = 1, 2.....p$ et les stocker dans un vecteur R.

(b) Initialisation de la sous histoire. Au debut de chaque sous histoire, posons $ip = 1$.

(c) Echantillonnage sans remise pendant la sous histoire :

1. Si $ip > p$ aller à (d).
2. Génerer aléatoirement un entier $iaux \in [ip, p]$.
3. Changer $r(ip)$ avec $r(iaux)$.
4. Générer une observation xd_i.
   Arréter s'il n'y a plus de demande de valeur et collecter les résultats finaux du dernier nombre premier utilisé et aller à (e).
5. Autrement, poser $ip = ip + 1$ et aller à $(c) - 1$

(d) Collecter les résultats aprés chaque sous histoire et aller à $(a) - 1$**.**

(e) Collecter les resultats aprés chaque histoire.

2.7 Conclusion

Dance ce chapitre, nous avons présenté quelques méthodes d'echantillonnages utilisées en simulation dont le but d'extraire des échantillons de variables aléatoires d'entrées du modèle de file d'attente M/M/1.

Chapitre 3

Etude de système de files d'attente M/M/1

3.1 Introduction

Dans ce chapitre, nous donnons quelques rappels sur le processus de naissance et de mort et le système de file d'attente de type M/M/1 [16].

3.2 Processus stochastique

Un processus stochastique $\{X(t)\}_{t\in T}$, est une fonction du temps dont la valeur à chaque instant dépend de l'issue d'une experience aléatoire . A chaque instant $t \in T$,X(t) est donc une variable aléatoire .
Un processus stochastique peut étre considéré comme une famille de variables aléatoires généralement non indépendantes.
L'ensemble des temps T peut *ê*tre discret ou continue.Ainsi X(t) définit l'état du processus à un instant donné t. L'ensemble noté E des valeurs que peut prendre le processus à chaque instant est appelé espace d'état et peut, de m*ê*me que T, étre discrect (fini ou infini) ou continu. En fonction des valeurs possible de T et E, on classifie les processus stochastique de la façon suivante :
1-Processus à temps discret et à espace d'état discret
2-Processus à temps continu et à espace d'état discret
3-Processus à temps discret et à espace d'état continu
4-Processus à temps continu et à espace d'état continu

3.3 Processus de naissance et de mort

Les phénoménes de naissance et de mort interviennent dans la modélisation des systèmes de files d'attente, et permettent de façon générale de décrire l'évolution temporelle de la taille d'une population d'un type donné (Biologie, démographie, Sociologie). Les processus de naissance et de mort $(N-M)$ sont des processus stochastiques à temps continu et à espace d'états discret. Ils sont markoviens (sans mémoire) et à partir d'un état n donné, les transitions ne sont possibles que vers l'un ou l'autre des états voisins. On parle alors de naissance et de mort [3].

Définition 3.1. *Soit* $\{X(t), t \geq 0\}$ *un processus stochastique, à espace d'états* $\mathbb{N} = 0, 1, ...$ *et homogène dans le temps :*

$\mathbb{P}(X(t+s) = i \mid X(s) = j) = P_{ij}(t)$*, ne dépend pas de s.*

$\{X(t), t \geq .0\}$ *est dit processus de naissance et de mort* $(N-M)$ *si les conditions suivantes sont satisfaites :*

- $P_{ii+1}(\Delta t) = \mathbb{P}(X(t+\Delta t) = i+1 \mid X(t) = i) = \lambda_i \Delta t + o(\Delta t), \quad i \geq 0.$
- $P_{ii-1}(\Delta t) = \mathbb{P}(X(t+\Delta t) = i-1 \mid X(t) = i) = \mu_i \Delta t + o(\Delta t), \quad i \geq 1.$
- $P_{ii}(\Delta t) = \mathbb{P}(X(t+\Delta t) = i \mid X(t) = i) = 1 - (\lambda_i + \mu_i)\,\Delta t + o(\Delta t), \quad i \geq 0.$

$\lambda_i > 0, \quad \mu_i > 0$ *sont les taux de transitions.*

λ_i : *taux de naissance (croissance).*

μ_i *: taux de mort (décroissance).*

Le générateur infinitésimal du processus est donc une matrice dite tridiagonale $A = (a_{ij})_{ij \in \mathbb{N}}$ *vérifiant* $a_{ij} = 0, |i-j| \geq 2$

$a_{nn+1} = \lambda_n, a_{nn-1} = \mu_n, \forall n \geq 1, (a_{01} = \lambda_0)$

$$A = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 & \dots \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & 0 & 0 & \dots \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & 0 & \dots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \end{pmatrix}$$

3.4 Processus de Poisson

Soit X_t le nombre de fois où se réalise un événement donné dans $[0,t], X_0 = 0$. $(X(t))_{t \geq 0}$ est dit processus de Poisson s'il satisfait aux trois conditions suivantes :

(C1) Le processus est homogene dans le temps c-à-d :

$$\mathbb{P}(X(t+s) - X(s) = k) = \mathbb{P}(X(t) = k) = P_k(t), \forall s > 0, t > 0, k \in \mathbb{N}.$$

Ceci veut dire que la probabilité d'avoir k événements dans un intervalle de longueur t ne dépend que de t et non pas de la position de l'intervalle par rapport à l'axe temporel.

(C2) Le processus et à accroissement indépendants, ceci veut dire que pour tout systeme d'intervalle disjoints, le nombre d'événements s'y produisant sont des variables aléatoires indépendantes en particulier :

$$\begin{aligned}\mathbb{P}(X(t+s)-X(s)=k, X(s)=j) &= \mathbb{P}(X(t+s)-X(s)=k)\mathbb{P}(X(s)=j)\\ &= P_k(t)P_j(s), \forall s>0, t>0\end{aligned}$$

(C3) La probabilité que deux événements ou plus se produisent dans un petit intervalle Δt est négligeable par rapport à la probabilité qu'il y ait un seul événememement :

$$P_k(\Delta t) = \begin{cases} o(\Delta t), & \text{si } k \geq 2 \\ \lambda \Delta t, & \text{si } k = 1 \\ 1 - \lambda \Delta t, & \text{si } k = 0 \end{cases}$$

λ est appele intensité du processus de Poisson (c'est le nombre moyen d'evénements par unité de temps).

Remarque 3.1. *Le processus de Poisson est un cas particulier du processus de naissance et de mort. A la différence de ces cas particuliers, les taux de transition dépendent généralement de l'état n dans lequel le système se trouve.*

3.5 Processus de Poisson et loi exponnentielle

3.5.1 Intervalle entre deux événements

Soit $\{N(t), t \geq 0\}$ un processus de Poisson de paramétre $\lambda > 0$,et T_n :la durée séparant le $(n-1)^{eme}$ évenement du n^{eme} évenement.

Théorème 4. *Les temps d'attente T_n d'un processus de poisson de paramétre λ, sont des v.a independantes identiquement distribuées selon une loi* $\exp \varepsilon(\lambda)$.

Théorème 5. *Le temps V separant un instant quelconque du prochain événement d'un processus de poisson est une variable aléatoire $\varepsilon(\lambda)$,distribuée selon une loi* $\exp(\lambda)$.

3.6 Régime transitoire et régime stationnaire

3.6.1 Régime transitoire

On cherche les probabilités d'état $P_n(t) = P(X(t) = n)$.
D'aprés la formule des probabilités totales, on a pour $n \geq 1$

$$\begin{aligned}
P_n(t+\Delta t) &= \mathbb{P}(X(t+\Delta t)=n) \\
&= \sum_{i=0}^{\infty} \mathbb{P}(X(t+\Delta t)=n \mid X(t)=i)\mathbb{P}(X(t)=i) \\
&= \sum_{i=0}^{\infty} P_i(t) P_{\text{in}}\,(\Delta t) \\
&= P_{n-1}(t)P_{n-1n}(\Delta t) + P_n(t)P_{nn}(\Delta t) + P_{n+1}(t)P_{n+1n}(\Delta t) + \\
&\sum_{i=n+2}^{\infty} P_i(t)P_{in}(\Delta t) \\
&= P_{n-1}(t)\lambda_{n-1}\Delta t + P_n(t)\left[1-(\lambda_n+\mu_n)\,\Delta t\right] + P_{n+1}(t)\mu_{n+1}\Delta t + o(\Delta t)
\end{aligned}$$

$$\frac{P_n(t+\Delta t)-P_n(t)}{\Delta t} = -\left(\lambda_n+\mu_n\right)P_n(t) + \lambda_{n-1}P_{n-1}(t) + \mu_{n+1}P_{n+1}(t) + \frac{o(\Delta t)}{\Delta t}$$

quand $\Delta t \longrightarrow 0$ on aura

$$P_n'(t) = -\left(\lambda_n+\mu_n\right)P_n(t) + \lambda_{n-1}P_{n-1}(t) + \mu_{n+1}P_{n+1}(t), \quad n \geq 1$$

de la même manière, on obtient pour $n=0$

$$P_0'(t) = -\lambda_0 P_0(t) + \mu_1 P_1(t).$$

Définition 3.2. *Les équations :*

$$\begin{cases} P_n'(t) = -\left(\lambda_n+\mu_n\right)P_n(t) + \lambda_{n-1}P_{n-1}(t) + \mu_{n+1}P_{n+1}(t), & n \geq 1 \\ P_0'(t) = -\lambda_0 P_0(t) + \mu_1 P_1(t), & n = 0 \end{cases}$$

sont appelées les équations différentielles de Chapmann Kolmogorov.

Si de plus, on connait les conditions initiales c-à-d $q_i = \mathbb{P}(X(0)=i)$, on détermine ainsi le régime transitoire du processus

$$P_n(t) = \sum_{i=0}^{\infty} q_i P_{in}(t)$$

Remarque 3.2. *Ces équations sont difficiles à résoudre , on ne considère alors que le régime stationnaire du processus.*

$$P_n = \lim_{t\to\infty} P_n(t) = \lim_{t\to\infty} \mathbb{P}(X(t)=n) = \mathbb{P}(X=n)$$

3.6.2 Régime stationnaire

Dans ce cas les équations de Kolmogorov deviennent : $\lim_{t\to\infty} P_n'(t) = 0, \forall n = 0,1,2\ldots.$

$$\begin{cases} \mu_1 P_1 = \lambda P_0, & n = 0 \\ \lambda_{n-1}P_{n-1} + \mu_{n+1}P_{n+1} = \left(\lambda_n+\mu_n\right)P_{n1} & n \geq 1 \end{cases}$$

Pour résoudre ce système d'équations linéaires, appelées, équations de balance, on additionne les $(n+1)$ premières équations, on trouve $\lambda_n P_n = \mu_{n+1} P_{n+1}, \quad \forall n \geq 0$, en admettant $\lambda_0 > 0$:

$$P_n = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} P_0, \quad n \geq 1$$

c-à-d

$$P_n = \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} P_0$$

Comme $\sum_{n=0}^{\infty} P_n = 1 \Longrightarrow P_0 + \sum_{n=1}^{\infty} \left(\prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} P_0 \right) = 1$, d'où

$$P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} \left(\prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} \right)}$$

3.7 Introduction aux files d'attente

L'origine des études sur les phénoménes d'attente remonte aux années 1909-1920 avec les travaux de A. K. Erlang concernant le réseau téléphonique de Copenhague. La théorie mathématique s'est ensuite développée noramment grace aux contributions de Palm, Kolmogorov, Khinchine, Pollaczeck, . . . et fait actuellement toujours l'objet de nombreuses publications scientifiques. Cette théorie s'est ensuite étendue à de nombreux champs d'application comme la gestion de stock, les télécommunications en général, la fiabilité des systèmes complexes, ...
Les phénoménes liés à l'attente dans un centre de service sont omniprésents dans notre société. Voici quelques exemples :

— Traffic urbain ou aérien.
— Réseaux téléphoniques.
— Circulation des piéces dans un atelier.
— Programme dans un système informatique.

On parle de phénoménes d'attente chaque fois que certaines unités appelés "Clients" se présente d'une maniére aléatoire à des "stations" afin de recevoir un service dont la durée est généralement aléatoire.

La théorie des files d'attente donne deux méthodes principales pour la résolution du conflit qui se produit lorsqu'un client arrive dans le système et trouve le(s) serveur(s) occupé(s) :

— Il quitte le système pour toujours sans étre servi, ce qui correspond au système d'Erlang avec refus (Erlang loss system) appelé aussi modèle d'appels perdus.
— Il peut attendre dans une file d'attente pour étre servi dés la libération du serveur,ce qui correspond au système de files d'attente classiques. (queueing systems).

3.7.1 Identification et représentation d'un système de file d'attente

On identifie un système de files d'attente par :

— La nature stochastique du processus des arrivées (définie par la distribution des intervalles séparant deux arrivées consécutives.)
— La distribution du temps de service.
— Le nombre de serveurs s.
— La capacité N du système. Si $N < \infty$, la file d'attente ne peut dépasser une longueur $N - s$ unités.
On suppose de plus, que toutes les variables aléatoires introduites pour décrire le système d'attente sont indépendantes.

3.7.2 Notations de Kendall

Pour spécifier le type de file d'attente qu'on étudie,on utilise habituellement la notation **A/B/s(N/k/DS)** ou :
A et **B** décrivent respectivement la distribution du temps intérarrivés et la distribution du temps du service .
s : le nombes de serveurs paralléles existant dans le système.
N : est la capacité du système(serveurs + file d'attente).
K :la popultalion des usagers.
DS :est la discipline de file d'attente.
Lorsque les trois derniers éléments de la notation de Kendall ne sont pas précisés,ils sont pris par défaut comme suite :

$$\mathbf{DS} = FIFO \ ,\mathbf{k} = \infty \ ,\mathbf{N} = \infty$$

3.7.3 Analyse mathématique d'une file d'attente

L'étude mathématiques d'un système de file d'attente se fait généralement par l'introduction d'un processus stochastique, défini de façon appropriée. On s'intéresse principalement au nombre de clients à l'instant $t = 0$.
En fonction des quantité qui définissent le système, on cherche à déterminer :
• Les probabilités d'état $P_n(t) = P(X(t) = n)$, qui définissent le régime transitoire du processus $\{X(t), t \geq 0\}$. Il est évident que les fonctions $P_n(t)$ dépendent de l'état initial ou la distribution initiale du processus.
•Le régime stationnaire du processus :

$$P_n = \lim_{t \to \infty} P_n(t) = P(X = n), n = 0, 1, 2, \ldots$$

3.7.4 Étude du système M/M/1

Ce système est les plus simple système, le flot des arrivées est poissonien de paramètre λ, la durée de service est exponentielle de paramètre μ, la capacité de la file est illimitée, il ya un seul serveur avec la discpline FIFO.
Considérons $\{X(t), t \geq 0\}$ le processus stochastique représentant le nombre de clients dans le système à l'instant t.
Grace aux propriétés fondamentales du processus de Poisson et la loi exponentielle on a :

- P(exactement 1 arrivée dans Δt) $= \lambda\Delta t + o(\Delta t)$.
- P(aucune arrivée dans Δt) $= (1 - \lambda\Delta t) + o(\Delta t)$.
- P(2 arrivées ou plus dans Δt) $= o(\Delta t)$.
- P(exactement 1 départ dans $\Delta t) \setminus (X(t) \geq 1) = \mu\Delta t) + o(\Delta t)$.
- P(aucun départ dans $\Delta t \setminus X(t) \geq 1) = (1 - \mu\Delta t) + o(\Delta t)$.
- P(2 départ dans $\Delta t \setminus X(t) \geq 1) = o(\Delta t)$.

Ces probabilités ne dépendent ni de t ni de l'état de $(X(t))$ ou il se trouve.
Soit $Pij(\Delta t) = P(X(t + \Delta t) = j \setminus X(t) = i), i, j = 0, 1, ...$ On a

$$\begin{aligned} P_{nn+1}(\Delta t) &= \lambda\Delta t(1 - \mu\Delta t) + o(\Delta t) \\ &= \lambda\Delta t + o(\Delta t) \left(\text{car } \lambda\mu(\Delta t)^2 = o(\Delta t)\right) \end{aligned}$$

$$\begin{aligned} P_{nn}(\Delta t) &= \lambda\Delta\mu\Delta + (1 - \lambda\Delta t)(1 - \mu\Delta t) + o(\Delta t)\forall n \geq 1 \\ &= 1 - (\lambda - \mu)\Delta t + o(\Delta t) \end{aligned}$$

$\mathbb{P}_{00}(\Delta t) = (1 - \lambda\Delta t) + o(\Delta t)$

$$\begin{aligned} P_{n+\ln}(\Delta t) &= (1 - \lambda)\mu\Delta t + o(\Delta t) \\ &= \mu\Delta t + \sigma(\Delta t)) \end{aligned}$$

Tandis que $\text{Pnm}(\Delta t) = o(\Delta t)$ pour $|m - n| \geq 2$
$\{X(t), t \geq 0\}$ est bien un processus de N − M de taux de transitions $\lambda_n = \lambda, \quad \forall n \geq 0$ et $\mu_n = \mu, \quad \forall n \geq 1$

Donc le système M/M/1 est une chaine de Markov continue de générateur infinitésimal (matrice des taux de transition)

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & 0 & \dots \\ \mu & -(\lambda + \mu) & \lambda & 0 & 0 & \dots \\ 0 & \mu & -(\lambda + \mu) & \lambda & 0 & \dots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \end{pmatrix}$$

3.7.5 Régime transitoire

Les equations de Kolmogorov sont données par :

$$\begin{cases} P_0'(t) = -\lambda P_0(t) + \mu P_1(t) \\ P_n'(t) = -(\lambda + \mu)P_n(t) + \lambda P_{n-1}(t) + \mu P_{n+1}(t), \quad n \geq 1 \end{cases}$$

Si l'on connait les conditions initiales du processus, on peut ainsi déterminer le régime transitoire du système d'attente M/M/1, c-à-d calculer $P_n(t)$ en fonction de λ et μ.

3.7.6 Régime stationnaire

On peut montrer que $\lim_{t\to\infty} P_n(t) = P_n$ existent et sont indépendantes de l'état initial du processus et que $\lim_{t\to\infty} P'_n(t) = 0, \forall n \geq 0$
En appliquant la formule de la distribution stationnaire d'un processus de naissance et de mort on a

$$P_n = P_0 \prod_{i=0}^{n-1} \frac{\lambda}{\mu}$$

D'ou

$$P_n = P_0 \left(\frac{\lambda}{\mu}\right)^n \cdot n \geq 0$$

On peut aussi obtenir à partir du graphe de transition les équations de balance (système d'équations linéaires homogènes) :

$$\begin{cases} \lambda P_0 = \mu P_1 \\ \lambda P_{n-1} + \mu P_{n+1} = (\lambda + \mu) P_n, \quad n \geq 1 \end{cases}$$

auquelles, on ajoute la condition $\sum_{n=0}^{\infty} P_n = 1$.
En additionnant les $(n+1)$ premieres équations, on trouve $\mu P_{n+1} = \lambda P_n$, d'où

$$P_n = P_0 \left(\frac{\lambda}{\mu}\right)^n \text{ et } P_0 = \frac{1}{\sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^n}$$

Si $\frac{\lambda}{\mu} < 1$ alors $P_0 = 1 - \frac{\lambda}{\mu}$, d'ou la distribution stationnaire de $M/M/1$ est

$$P_n = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^n, \forall n \in \mathbb{N}$$

En posant $\rho = \frac{\lambda}{\mu}$, l'équation devient

$$P_n = (1 - \rho)\rho^n \quad \forall n \in \mathbb{N}$$

Remarque 3.3.
1. *Le régime stationnaire du système M/M/1 est donc gouverné par la loi géométrique sur N de paramètre* $1 - \rho$.
2. ρ *est appelé coefficient d'utilisation du système, ou intensité de traffic correspondant. II constitue une mesure de saturation du système.*
3. $\rho = \frac{\lambda}{\mu}$ = *nombre moyen d'arrivées* $\times$ *durée moyenne de service.*
4. $\rho = 1 - P_0$ *est la probabilité d'occupation de la station service.*
5. $\lambda < \mu$ *est la condition nécessire et suffisante pour lergodicité du système M/M/1.*
6. *si* $\lambda \geq \mu, \lim_{t\to\infty} P_n(t) = P_n = 0, \quad \forall n \geq 0$ *(la longneur de la file dépasse toute limite).*

Théorème 6. *Si* $\lambda < \mu$, *au régime stationnaire, le processus des départs pour le système M/M/1 est un processus de Poissan de paramétre* λ.

3.8 Caractéristiques d'une file d'attente M/M/1

- Le nombre moyen de clients dans le système :

  soit X le nombre de clients se trouvant dans le système et soit L le nombre moyen de clients dans le système .

$$\begin{aligned} L = \mathbb{E}(X) &= \sum_{n=0}^{\infty} nP_n \\ &= (1-\rho)\sum_{n-0}^{\infty} n\rho^n \\ &= (1-\rho)\rho\frac{1}{(1-\rho)^2} \end{aligned}$$

  D'où

$$L = \frac{p}{1-\rho} = \frac{\lambda}{\mu-\lambda}$$

- Le nombre moyen de clients dans la file :

  soit L_q le nombre moyen de clients dans la file et soit X_q le nombre de clients se trouvant dans la file d'attente, avec

$$X_q = \begin{cases} 0 & \text{si } X = 0, \\ X-1 & \text{si } X \geq 1. \end{cases}$$

$$\begin{aligned} L_q = \mathbb{E}\left(X_q\right) &= \sum_{n=0}^{\infty}(n-1)P_n \\ &= (n-1)(1-\rho)\rho^n \\ &= (1-p)\left[\sum_{n=1}^{\infty} n\rho^n - \sum_{n=1}^{\infty}\rho^n\right] \\ &= \frac{\rho}{1-\rho} - p \end{aligned}$$

  Donc le nombre moyen de clients dans la file est :

$$L_q = \frac{\rho^2}{1-\rho} = \frac{\lambda^2}{\mu(\mu-\lambda)}$$

Formules de Little

Soit, W le temps moyen de séjour d'un client dans le système et W_q temps d'attente moyen d'un client. En utilisant les formules de Little on aura :

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

et

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

3.9 Conclusion

Ce chapitre est consacré pour l'étude du système de files d'attente M/M/1 dans le but d'évaluer sa performances à travers la simulation.

Chapitre 4

Application

4.1 Introduction

Après avoir introduit dans les chapitres précédents, la méthode d'échantillonnage descriptif amélioré et le système de file d'attente de type M/M/1, dans ce chapitre nous nous intéressons à la modélisation et à la simulation d'un système de file d'attente de type M/M/1 par l'application de la méthode d'échantillonnage descriptif amélioré. Dans un premier temps, nous abordons la modélisation d'un système à événements discrets à savoir une file d'attente de type M/M/1. Par la suite, nous donnons les différents algorithmes et organigrammes qui vont nous permettre d'implémenter la méthode d'EDA.

4.2 Modélisation d'une file d'attente

Une file d'attente est modélisée de la manière suivante :

1. Les temps entre deux arrivées successives des clients sont des variables aléatoires indépendantes et de même loi. Si les instants d'arrivées forment un processus de poisson, ce processus sera représenté par la lettre M (Markovien), si ce processus est quelconque, il sera noté G (général).
2. Les temps de service aux guichets ou durées des requêtes sont des variables aléatoires indépendantes et de même loi. Si les durées de service suivent une loi exponentielle, cette loi est représentée par la lettre M (Markovien), si le processus de service est quelconque, il sera noté :(D déterministe), E_k (Erlang d'ordre K) ou G (général).
3. Les nombres de guichets ou serveurs noté S.
4. Eventuellement, la taille de la file d'attente ou de la salle d'attente peut être limitée. Par exemple pour les réseaux téléphoniques, si tous les serveurs sont

occupés (réseau saturé) les appeles sont refusés.La longueur de la salle d'attente est donc nulle. Enfin la salle d'attente est commune à tous les serveurs.

Le modèle de files d'attente retenu pour l'implémentation des différents algorithmes est le modèle M/M/1.

4.3 Modèle de simulation d'une file d'attente M/M/1

Les caractéristiques d'une file d'attente à un serveur sont :

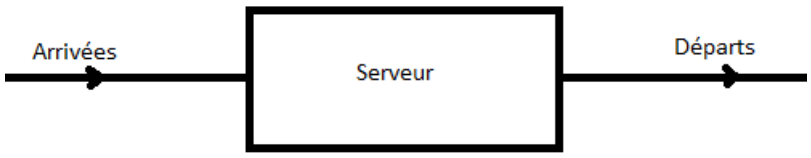


FIGURE 4.1 – Système de fille d'attente à un serveur

1. Le taux d'arrivée de clients.
2. La discipline de la file d'attente.
3. La distribution de temps de service.

Si un client arrive et que le serveur est inoccupé, il y va directement, s'il y a une file d'attente ou le serveur est occupé, le client entre en attente, se plaçant dans la file à une position déterminée par la discipline. Quand un client arrive à la station de service, il est servi en un temps determiné par le taux de service, le service termine, le client servi sort et un nouveau client vient s'il reste des clients en attente, sinon le serveur devient inoccupé.

Nous présentons dans ce qui suit un modèle général de simulation :

a) Les composantes :

Serveur, station de service, file d'attente, clients.

b)Les variables :

- Exogènes (indépendantes)
  - Distribution de probabilité des temps inter arrivées (taux).

— Discipline de la file d'attente (ou relation fonctionnelle).
— Distribution de probabilité des temps de service.

◦ Exogènes (dépendantes)
— FA : nombre de clients en file d'attente.
— TA : intervalle de temps jusqu'à la prochaine arrivée.
— TPA : temps d'arrivée de prochain client.
— TS : temps de service de l'actuel client.
— TAT : temps d'attente total.
— TLT : temps libre total du serveur (durée du répit).

4.3.1 Organigramme du modèle de file d'attente à 1 serveur

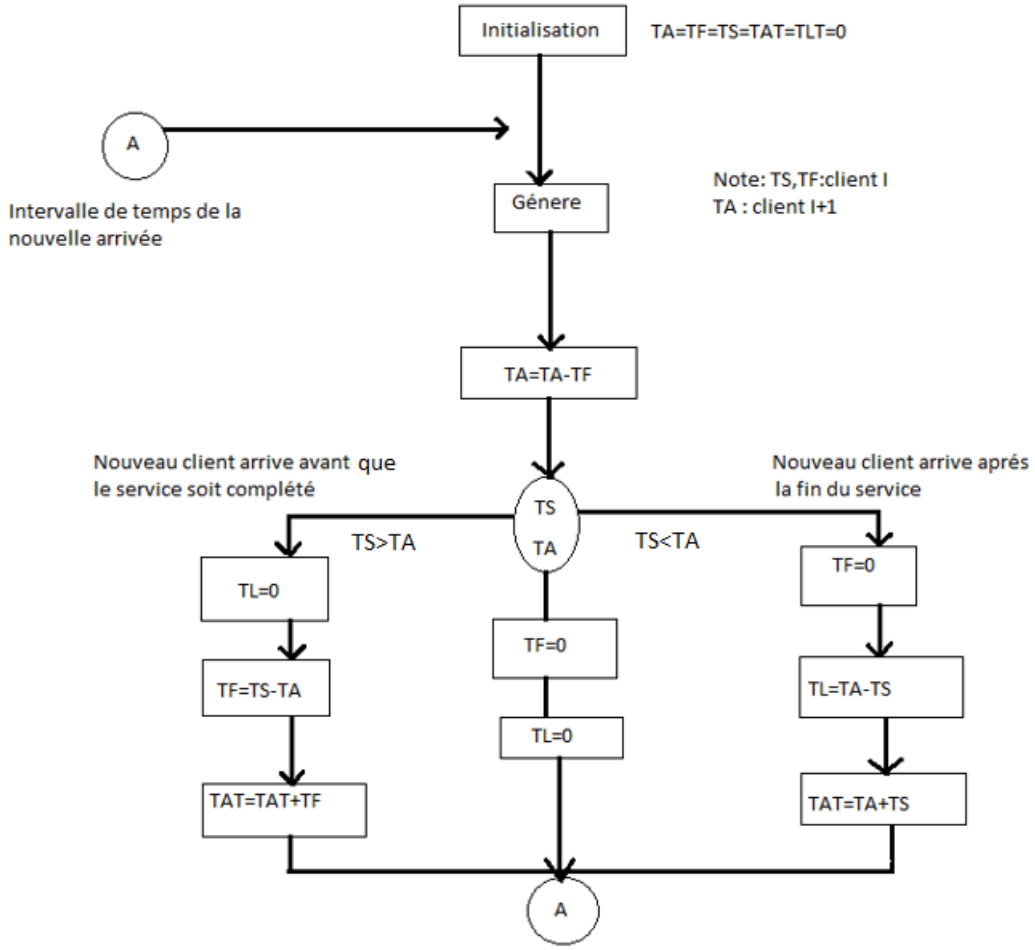


FIGURE 4.2 – Organigramme du modèle de file d'attente à 1 serveur

— TF :intervalle de temps que le client doit passer en file d'attente avant d'être servi.

— TL :intervalle de temps pendant lequel le serveur est libre et attend pour la prochaine.

4.4 Organigrammes de l'ensemble des fonctions constituant la méthode EDA

4.4.1 Organigramme de la fonction $alea\_min\_max(MIN, MAX)$

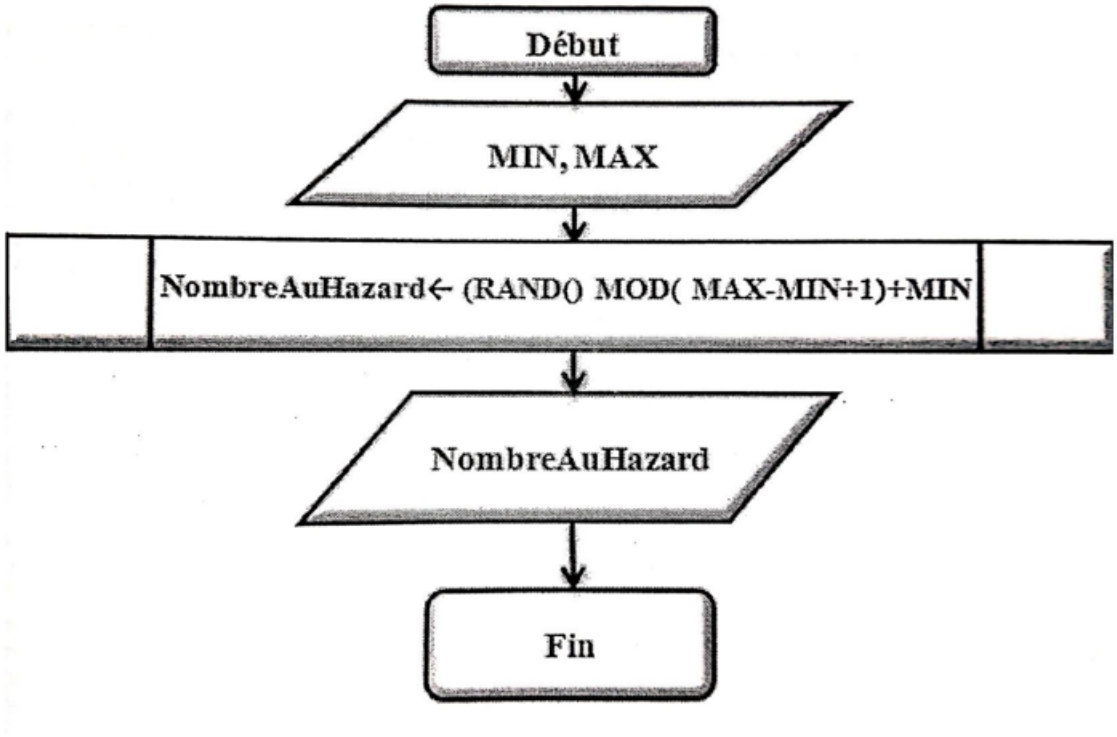
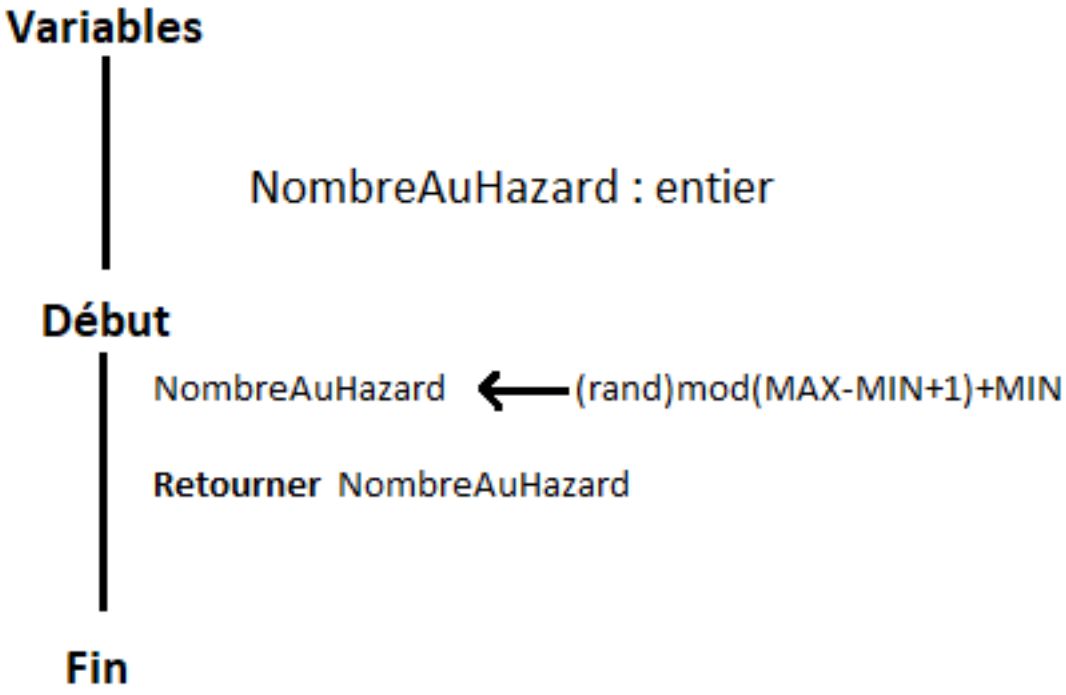


FIGURE 4.3 – Organigramme de la fonction $alea\_min\_max$

L'Algorithme correspondant est :

Algorithme 1 *Fonction* : *alea_min_max*
Fonction *alea_min_max* (MIN,MAX :entier) :entier



4.4.2 Organigramme de la fonction *est_premier*(N)

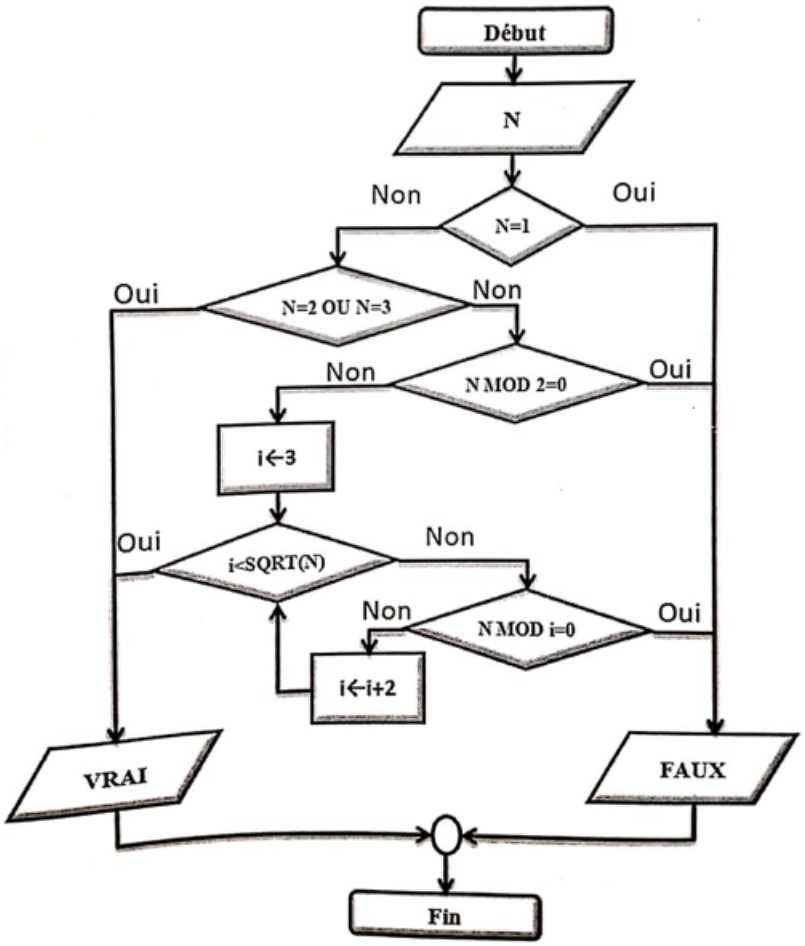
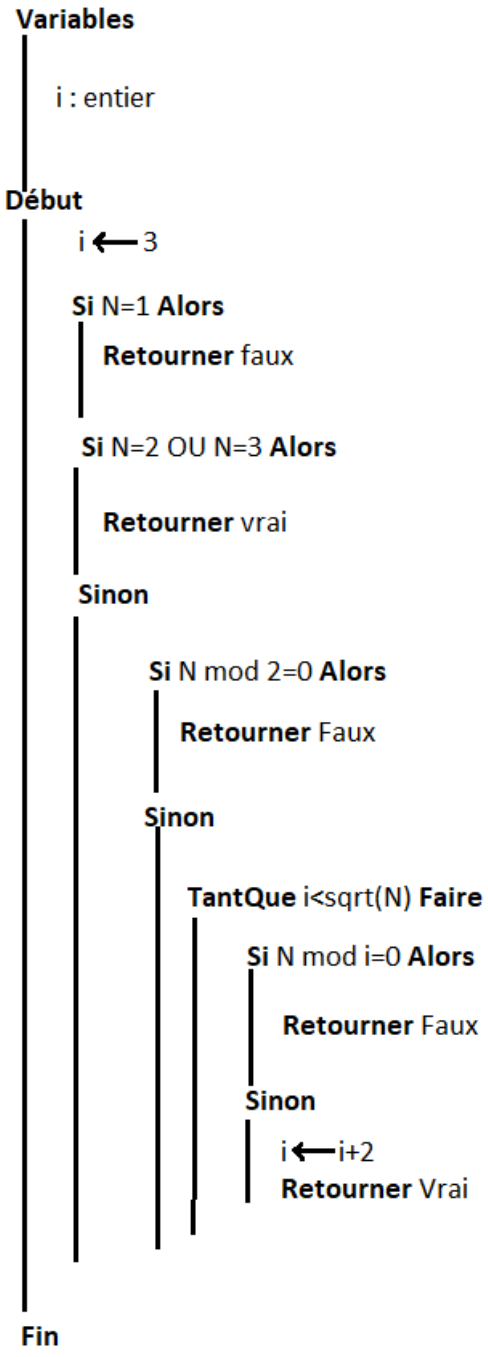


FIGURE 4.4 – Organigramme de la fonction *est_premier*(N)

L'algorithme correspondant est :

Algorithme 2 *Fonction Est_premier*
Fonction : *Est_premier*(N :entier) :booléen



4.4.3 Organigramme de la fonction pgetrand()

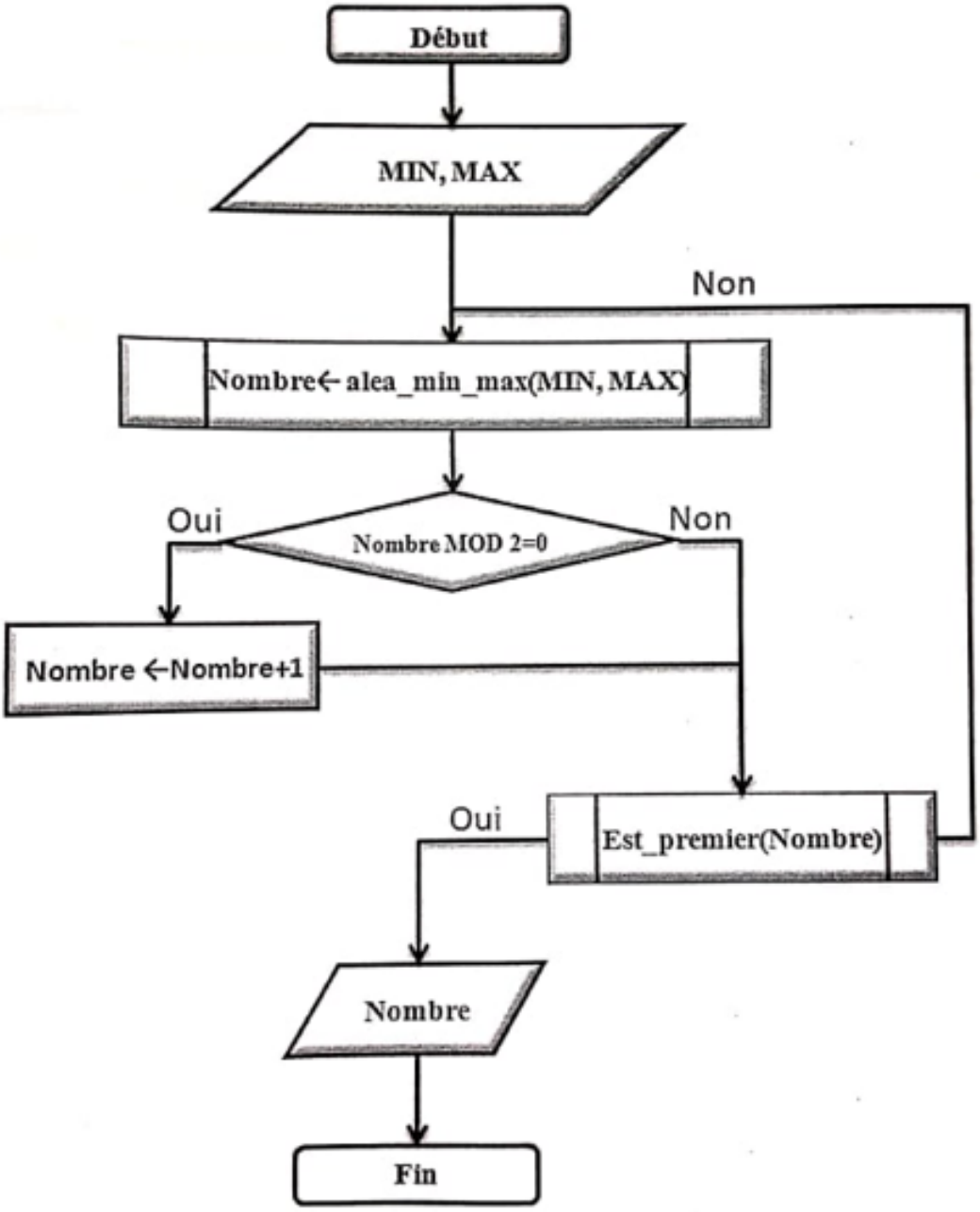
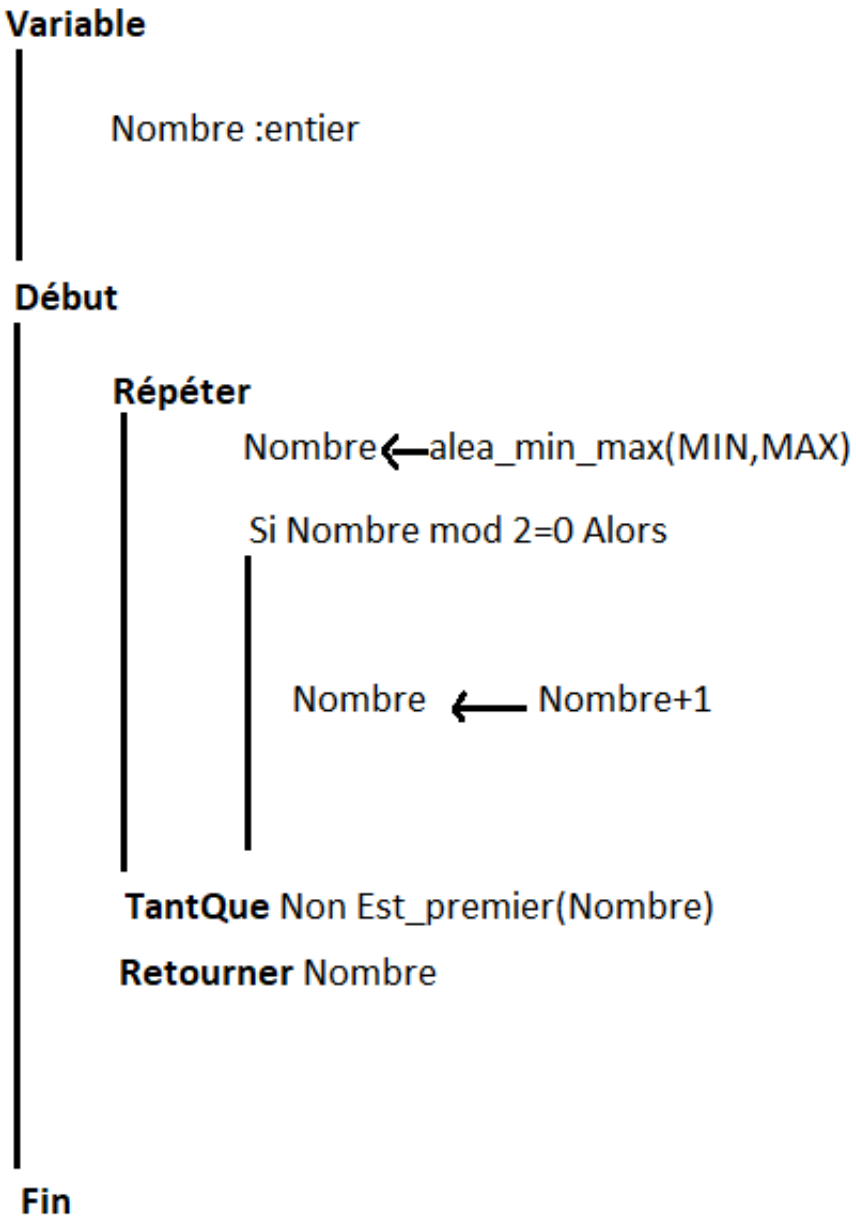


FIGURE 4.5 – Organigramme de la fonction pgetrand

L'Algorithme correspondant est :

Algorithme 3 *Fonction pgetrand*

Fonction : pgetrand(MIN,MAX :entier) :entier



4.4.4 La fonction ugetrand()

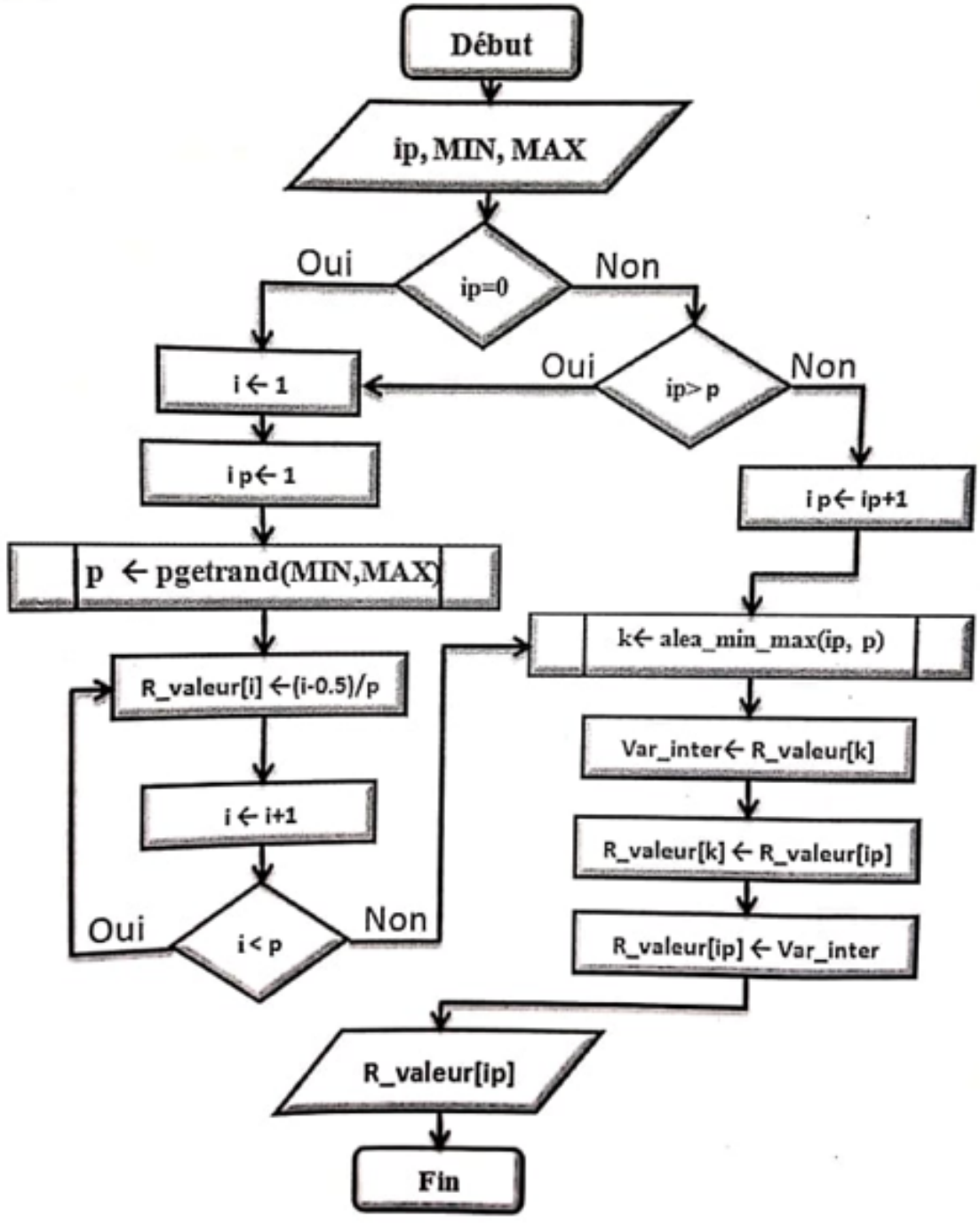


FIGURE 4.6 – Organigramme de la fonction ugetrand

L'Algorithme correspondant est :

Algorithme 4 *Fonction ugetrand*

```
Variable
    Nbre_premier,ip:entier
    R_valeur:Tableau réel[nbre_premier]

Fonction ugetrand(MIN,MAX:entier):réel

Variable
    i,k : entier
    var_inter :réel

Début
    Si ip=0 Alors
        Etiquette a: ip ←1
        nbre_premier← pgetrand(MIN,MAX)
        Pour i variant De 1 à nbre_premier Faire
            R_valeur[i] ←(i-0.5)/nbre_premier
        k←alea_min_max(ip,nbre_premier)
        var_inter ←R_valeur[ip]
        R_valeur[ip] ←R_valeur[K]
        R_valeur[K] ←var_inter

        Retourner R_valeur[ip]
    Sinon

        ip ←ip+1

        Si ip>nbre_premier Alors

            Aller à Etiquette a

        Sinon

            k←alea_min_max(ip,nbre_premier)
            var_inter←R_valeur[ip]
            R_valeur[ip]←R_valeur[K]
            R_valeur[K] ←var_inter

            Retourner R_valeur[ip]

Fin
```

4.4.5 Description

— La fonction **alea_min_ max()** renvoie une valeur entière positives aléatoire comprise entre les valeurs **MIN et MAX**. Ces valeurs seront introduites par l'utilisateur.
— La fonction **Est_premier()** vérifie si le nombre N est premier ou non. Si oui elle retourne la valeur 1 , sinon la valeur retournée c'est 0.
— La fonction **pgetrand()** renvoie aléatoirement une valeur entière positives première comprise entre les valeurs **MIN et MAX**. Ces valeurs sont fixées par l'utilisateur.
— La fonction **ugetrand()** renvoie une valeur réelle aléatoire comprise entre 0 et 1 en fonction du nombre premier qui est généré aléatoirement par la fonction **pgetrand()** : ce nombre est compris entre **MIN, et MAX**.

Erreurs

Les fonctions *alea_min_max*(), *pgetrand*() et *ugetrand*() peuvent déclencher les erreurs suivantes :
- **Si (MIN** < 0 **ou MAX** < 0 **ou MAX** $<$ **MIN) alors**
Elles retournent : Erreur Paramètres d'entrées, vérifiez MIN et MAX
- Si (**MAX** > 429496700) alors
Elles retournent : Valeur **MAX** assez grande !
Dans les deux cas, l'exécution du programme s'arrête.

4.4.6 Notes

1. Pour utiliser les fonctions de la bibliothéque EDA, il ne faut pas oublier d'ajouter l'argument-Im sur la ligne de commande gcc,lors de l'édition des liens.
2. Pour la méthode de monte carlo, il suffit d'appeler la fonction **rand**() sous n'importe quel language de programmation.

4.5 Résultats numériques

Une file d'attente stationnaire M/M/1 sous la discpline FIFO, avec des arrivées poissonnieres de paramétre λ et une distribution de service exp de paramétre μ a été simulée sous linux utilisant le language de programmation C.
La méthode d'échantillonnage descriptif amélioré est utilisée pour générer les échantillons des variables d'entrées pour observer des échantillons des variables de sortie , Le temps moyen d'attente dans le systéme W, et Le temps moyen d'attente dans la file W_q
Trois expériences de simulation ont été realisées avec 100 réplications dont chaque experience est testée trois fois, les paramétres λ et μ sont choisis par l'utilisateur de sorte que $\lambda < \mu$ (condition de stabilité du systéme). Les résultats des trois expériences sont donnés dans les tableaux suivants :

Valeurs théoriques		Résultats obtenues pour $\lambda = 3$ et $\mu = 5$		
		test1	test2	test3
W_q	0.3	0.28	0.309	0.284
W	0.5	0.472	0.502	0.477
$\frac{1}{\lambda}$	0.33333	0.334	0.33	0.332

TABLE 4.1 – Simulation d'une file d'attente M/M/1 pour $\rho = 0.6$.

Valeurs théoriques		Résultats obtenues pour $\lambda = 2$ et $\mu = 2.6667$		
		test1	test2	test3
W_q	1.1249	1.0105	1.1308	1.2297
W	1.4999	1.3971	1.6201	1.6589
$\frac{1}{\lambda}$	0.5	0.4832	0.5105	0.5132

TABLE 4.2 – Simulation d'une file d'attente M/M/1 pour $\rho = 0.75$.

Valeurs théoriques		Résultats obtenues pour $\lambda = 0.9$ et $\mu = 1$		
		test1	test2	test3
W_q	9.00	8.4532	8.6666	8.3120
W	10	9.1593	9.3804	9.1532
$\frac{1}{\lambda}$	1.1111	1.0032	0.9972	0.9007

TABLE 4.3 – Simulation d'une file d'attente M/M/1 pour $\rho = 0.9$.

4.5.1 Interprétation des résultats

Les trois tableaux montrent que les résultats empériques obtenus par l'EDA sont proche des valeurs théoriques pour différentes valeurs de ρ a savoir $\rho = 0.6$, $\rho = 0.75$ et $\rho = 0.9$.

Conclusion Générale

Conclusion

Dans ce mémoire nous avons commencé par introduire la simulation à événements discrets utilisée dans notre application. La génération d'échantillons suivant différéntes lois de probabilités usuelles a été aussi introduite selon différentes méthode de simulation (inversion et rejet). Ensuite, les processus stochastiques et la file d'attente M/M/1 ont été étudiés pour permettre la validation de notre application.

La méthode de Monte-Carlo a été étudiée avec ses avantages et ses inconvénients. Pour pallier au problème de précision des estimateurs obtenus par MC, nous avons présenté les différentes techniques de réduction de la variance ainsi que quelques méthodes d'échantillonnages qui réduisent la variance, particuliérement l'EDA qui fut séléctionné dans des travaux récents, comme une meilleure méthode comparée à l'échantillonnage aléatoire(l'EA) et l'échantillonnage descriptif (l'ED).

Pour pouvoir utiliser l'EDA un composant logiciel a été développé qui est une implémentation de la méthode EDA. Ce composant a été développé en utilisant le language de programmation C sous le système d'exploitation Linux.En particulier, les differentes structures de ce composant et les algorithmes ont été données. Le manuel et un exemple d'utilisation du composant a été aussi présenté dans ce mémoire pour faciliter la tache a l'utilisateur.

Finalement, la méthode EDA a été validée à travers le modèle de file d'attente M/M/1. Les résultats obtenus à travers la simulation sont très proche des valeurs théoriques.

Perspectives

Les résultats obtenus dans le cadre de ce travail ouvrent de nombreuses perspectives :

- Application du générateur de l'EDA à différentes systèmes de files d'attente ainsi que dans d'autres domaines,l'ingénieurie mathématiques, la cryptographie etc....
- Comparaison de l'EDA avec d'autres générateurs de nombres pseudo-aléatoire.
- Généralisation de la méthode d'échantillonnage descriptif amélioré à des dimensions supérieurs.

Bibliographie

[1] Aloui, A and Ourbih–Tari, M. The use of Refined Descriptive Sampling and applications in parallel Monte Carlo Simulation. Computing and informatics. Vol 30, n 4, (2011), pp 681-700.

[2] Baiche, L and Ourbih-Tari, M. Large sample variance of simulation using refined descriptive sampling : Case of independent variables. Accepted paper in Communications in Statistics : Theory and Methods. Taylor and Francis. Accepté le 03/12/2014.

[3] Bouraine, L. cours de processus stochastique, Département de mathématiques, université de Abderahmane Mira Béjaia ,2020-2021.

[4] Fishman, G. S. Monte Carlo : Concepts, algorithms and applications, Springer-Verlag, 1997.

[5] Hickernell, F, G, Hong, H, S, L'ecuyer, P and Lemieux, C. Extensible lattice sequences for quasi Monte Carlo quadrature, SIAM Journal on scientific computing. 22, 3 (2000) 1117-1138.

[6] James, A. B, Variances reduction techniques, J. opl. Res. Soc, Vol 36, (1985) 525-530.

[7] Karsten Decker, M. The Monte Carlo method in science and engineering : Theory and application. Computer Methods in Applied Mechanics and Engineering, August 1991.

[8] Makoto, M and Takuji, N. Creation of pseudorandom number generators. Monte Carlo and Quasi Monte Carlo Methods, (1998).

[9] Niederreiter, H and Spanier, J. Monte Carlo and Quasi Monte Carlo Methods, P : 56-69 berlin, Springer (2000).

[10] Ourbih-Baghdali, L, Ourbih–Tari, M and Dahmani, A. A pseudo random number generator under Windows using Refined Descriptive Sampling. Computer Technology and Application. Vol 4, n°2, (2013) pp 85-92.

[11] Ourbih-Tari, M and Dahmani, A. Refined Descriptif Sampling : A better approach to Monte Carlo simulation, SIMPAT 428 21- 22.

[12] Robert, C and Cassella, G. Monte Carlo statistical methods, 2nd edn. Springer, New York (2004).

[13] Robert, P. Réseaux et files d'attente, Mathématique et Applications.

[14] Saliby, E. A reappraisal of some simulation fundamentals, Ph. D Thesis, Departement of Opertional Research, Université of Lancaster 1980.

[15] Saliby, E. Descriptive Sampling : A better approach to Monte Carlo simulation, journal of the Operational Research Society. 42, 12 (1990) 1133- 1142.

[16] Saliby, E. Understanding the variability of simulation results : An empirical study, journal of the Operational Research society. 41, 12 (1990) 1133- 1142.

[17] Tuffin, B and Leny, L. M. Parallélisation d'une combinaison des méthodes de Monte Carlo et quasi Monte Carlo et application aux réseaux de files d'attente, RAIRO Operations Research. 34 (2000) 85-98.