

# Simulations et modèles

Fabian Bastin

Année académique 2012-2013



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Concepts de base . . . . .	1
1.1.1	Pourquoi simuler ? . . . . .	2
1.1.2	Modèles . . . . .	3
1.2	Logiciels de simulation . . . . .	6
1.3	Applications . . . . .	8
1.4	Exercices . . . . .	8
<b>2</b>	<b>Modélisation : principes de base</b>	<b>11</b>
2.1	Introduction et définitions . . . . .	11
2.2	Techniques de validation . . . . .	12
2.3	Qualité des données . . . . .	13
2.3.1	Hypothèses simplificatrices . . . . .	14
2.3.2	Robustesse. . . . .	15
2.4	Types de méthodes pour choisir les lois . . . . .	16
2.4.1	Approche paramétrique . . . . .	17
2.4.2	Estimation de paramètres . . . . .	19
2.5	Tests d'Ajustement . . . . .	22

2.5.1	Procédures heuristiques . . . . .	23
2.6	Exercices . . . . .	24
<b>3</b>	<b>Simulation à événements discrets</b>	<b>25</b>
3.1	Approche par événements . . . . .	25
3.2	Le temps de simulation . . . . .	27
3.2.1	Simulation sans liste d'événements. . . . .	31
3.3	Approche par processus . . . . .	32
3.4	Exercices . . . . .	37
<b>4</b>	<b>Générateurs uniformes <math>[0, 1]</math></b>	<b>39</b>
4.1	Principes de base . . . . .	39
4.2	Générateurs récursifs . . . . .	40
4.2.1	Générateur congruentiel linéaire (GCL) . . . . .	40
4.2.2	Implantation de générateurs congruentiels linéaires . . . . .	41
4.2.3	Générateur récursif multiple (MRG) . . . . .	44
4.3	Générateurs à sous-suites multiples . . . . .	46
4.4	Structure de réseau . . . . .	49
4.4.1	Indices lacunaires. . . . .	53
4.4.2	Mesures d'uniformité. . . . .	53
4.5	Autres bornes. . . . .	54
4.5.1	MRGs combinés . . . . .	55
4.6	Générateurs basés sur des récurrences linéaires dans $\mathcal{F}_2$ . . . . .	56
4.6.1	Générateur de Tausworthe (ou LFSR : linear feedback shift register) . . . . .	57
4.6.2	Generalized feedback shift register (GFSR) . . . . .	59

4.6.3	Twisted GFSR . . . . .	60
4.7	Générateurs combinés sur $\mathcal{F}_2$ . . . . .	61
4.8	Tests statistiques . . . . .	62
4.8.1	Test de collisions . . . . .	63
4.8.2	Espacement entre anniversaires . . . . .	64
4.8.3	Test d'autocorrélation . . . . .	64
4.9	Quelques générateurs largement utilisés . . . . .	65
4.10	Résultats de tests . . . . .	65
4.10.1	Test de collisions . . . . .	65
4.10.2	Espacements d'anniversaire . . . . .	66
4.11	Tests systématiques pour des familles de générateurs de nombres aléatoires . . . . .	67
4.11.1	Librairie de tests . . . . .	67
4.12	Exercices . . . . .	68
<b>5</b>	<b>Génération de variables aléatoires non uniformes</b>	<b>69</b>
5.1	Motivations . . . . .	69
5.2	Inversion . . . . .	70
5.2.1	Implantation de l'inversion pour les lois discrètes . . . . .	72
5.2.2	Inversion pour l'interpolation linéaire . . . . .	73
5.2.3	Processus de Poisson stationnaire . . . . .	73
5.3	Autres approches . . . . .	74
5.3.1	Composition . . . . .	74
5.3.2	Convolution . . . . .	74
5.4	Acceptation-rejet . . . . .	75
5.5	Cas particuliers . . . . .	78

5.5.1	Méthode de Box-Muller pour la loi normale . . . . .	78
5.5.2	Variable aléatoire de Poisson . . . . .	80
5.5.3	Uniforme sur la sphère unité . . . . .	80
5.6	Exercices . . . . .	82
<b>6</b>	<b>Estimation et intervalles de confiance</b>	<b>83</b>
6.1	Estimateurs statistiques . . . . .	83
6.2	Efficacité des estimateurs . . . . .	84
6.3	Intervalles de confiance . . . . .	85
6.4	Horizon fini . . . . .	86
6.4.1	Approximation normale . . . . .	88
6.5	Intervalle de confiance pour une loi discrète . . . . .	89
6.6	Estimation séquentielle . . . . .	90
6.6.1	Procédure à deux étapes . . . . .	91
6.6.2	Estimation séquentielle . . . . .	91
6.7	Estimation de quantiles . . . . .	91
6.8	intervalle de confiance par rééchantillonnage (“bootstrap”) . . . . .	93
6.8.1	Principe de plug-in . . . . .	93
6.8.2	Bootstrap non-paramétrique . . . . .	94
6.8.3	Bootstrap non-paramétrique de base . . . . .	94
6.8.4	Bootstrap-t non-paramétrique . . . . .	95
6.8.5	Estimation du biais . . . . .	96
6.9	Intervalle de confiance pour une fonction de plusieurs moyennes . . . . .	97
6.9.1	Différence entre deux moyennes . . . . .	100
6.10	Moyenne sur horizon infini, état stationnaire . . . . .	101
6.10.1	Détection et réduction du biais initial . . . . .	102

6.11	Horizon tronqué . . . . .	104
6.11.1	Intervalle basé sur une simulation unique . . . . .	106
6.11.2	Coûts (ou revenus) actualisés . . . . .	109
6.12	Résumé du chapitre . . . . .	109
6.13	Exercices . . . . .	110
<b>7</b>	<b>Amélioration de l'efficacité</b>	<b>113</b>
7.1	Exemple introductif . . . . .	113
7.1.1	Estimation indirecte. . . . .	114
7.1.2	Variable de contrôle (VC). . . . .	115
7.1.3	Stratification. . . . .	116
7.1.4	Stratégies combinées. . . . .	117
7.1.5	Résultats numériques pour $n = 1000$ . . . . .	118
7.1.6	Comparaison de deux systèmes similaires . . . . .	118
7.2	Induction de corrélation . . . . .	119
7.3	Valeurs aléatoires communes (VAC) . . . . .	120
7.4	Variables de contrôle (VC) . . . . .	121
7.4.1	Types de VCs . . . . .	122
7.4.2	Estimation de $\beta^*$ : propriétés asymptotiques . . . . .	123
7.4.3	Cas multinormal . . . . .	123
7.4.4	Expériences pilotes pour estimer $\beta^*$ ? . . . . .	124
7.5	Espérance conditionnelle (EC) ou Monte Carlo conditionnel (CMC) . . . . .	125
7.5.1	Cadre général . . . . .	125
7.6	Stratification . . . . .	127
7.7	Variables antithétiques (AV) . . . . .	129
7.7.1	Méthode AV générale . . . . .	130

7.7.2	Méthode AV “classique” : Paires antithétiques. . . . .	130
7.7.3	Combinaison VAC-AV . . . . .	130
7.8	Méthodes QMCR . . . . .	131
7.8.1	Techniques quasi-Monte Carlo . . . . .	132
7.8.2	Perturbations aléatoires . . . . .	133
7.9	Exercices . . . . .	134
<b>A</b>	<b>Rappels de Statistiques et de Probabilités</b>	<b>135</b>
A.1	Variables aléatoires . . . . .	135
A.2	Quelques caractérisations des distributions. . . . .	136
A.2.1	Variables discrètes et continues . . . . .	136
A.2.2	Fonction de répartition . . . . .	137
A.2.3	Propriétés . . . . .	138
A.2.4	Convergence de variables aléatoires . . . . .	138
A.3	Quelques lois de probabilité discrètes . . . . .	139
A.3.1	Loi uniforme discrète . . . . .	139
A.3.2	Loi binomiale . . . . .	139
A.3.3	Loi multinomiale . . . . .	139
A.3.4	Loi géométrique . . . . .	140
A.3.5	Loi de Poisson . . . . .	141
A.4	Lois continues univariées . . . . .	141
A.4.1	Loi uniforme . . . . .	141
A.4.2	Loi triangulaire . . . . .	142
A.4.3	Loi normale . . . . .	142
A.4.4	Loi lognormale . . . . .	144
A.4.5	Loi de Erlang . . . . .	144

A.4.6	Loi exponentielle . . . . .	145
A.4.7	Loi de Cauchy . . . . .	146
A.4.8	Loi de Weibull . . . . .	147
A.4.9	Loi de Gumbel . . . . .	147
A.4.10	Loi beta . . . . .	148
A.4.11	Loi du chi-deux . . . . .	149
A.4.12	Familles de Johnson . . . . .	150
A.5	Lois tronquées . . . . .	151
A.6	Lois décalées . . . . .	151
A.7	Mélanges de lois . . . . .	151
A.8	Lois multivariées . . . . .	152
A.8.1	Loi normale multivariée . . . . .	152
A.8.2	Indépendance, covariance, corrélation . . . . .	153
A.8.3	Bornes de Fréchet . . . . .	154
A.8.4	Autres mesures de dépendance . . . . .	155
A.8.5	Dépendance vs corrélation. . . . .	155
A.9	Fonctions de dépendance (copules) . . . . .	156
A.10	Lois empiriques et quasi-empiriques . . . . .	156
A.11	Processus stochastiques . . . . .	159
A.11.1	Processus markoviens . . . . .	160
A.11.2	Filtrages et temps d'arrêt . . . . .	160
A.11.3	Processus de Poisson . . . . .	161
A.11.4	Processus brownien . . . . .	163
A.11.5	Estimation de moyennes, variances, et corrélations . . . . .	165
A.11.6	Convergence . . . . .	166

A.12 Illustration avec SSJ . . . . .	167
A.13 Tests d'hypothèse . . . . .	168
A.13.1 Test d'adéquation . . . . .	168
A.14 Exercices . . . . .	169
<b>B Rappels de géométrie et d'algèbre</b>	<b>171</b>
B.1 Corps . . . . .	171
B.2 Hyperplan . . . . .	171
<b>C Systèmes de file d'attente</b>	<b>173</b>
C.1 Notations pour les systèmes de files d'attente . . . . .	174
C.2 Mesures de performance pour les files d'attente . . . . .	175
<b>Bibliographie</b>	<b>176</b>
<b>Index</b>	<b>179</b>

# Table des figures

1.1	Capture d'écran du simulateur open source FlightGear Flight Simulator . . . . .	3
1.2	Types de modèles . . . . .	4
1.3	Organisation des notes de cours . . . . .	5
1.4	Capture d'écran du micro-simulateur Paramics . . . . .	7
3.1	Représentation schématique d'une simulation à événements discrets	26
3.2	Schéma d'une simulation à événements discrets. . . . .	28
3.3	Approche par processus . . . . .	34
4.1	Comportement d'un générateur de nombres aléatoires . . . . .	39
4.2	Décomposition de l'entier $x$ . . . . .	43
4.3	Sous-suites d'un générateur de nombres aléatoires. . . . .	47
4.4	distribution de Cauchy erronément générée . . . . .	47
4.5	LCG avec $m = 101$ et $a = 12$ ; $\mathbf{v}_1 = (1, 12)/101$ , $\mathbf{v}_2 = (0, 1)$ . .	51
4.6	LCG with $m = 101$ and $a = 7$ . . . . .	52
4.7	LCG with $m = 101$ and $a = 51$ . . . . .	53
4.8	Registre de décalage à feedback linéaire. . . . .	58

4.9	The XKCD random numbers generator ( <a href="http://xkcd.com/c221.html">http://xkcd.com/c221.html</a> ). . . . .	63
5.1	Majoration de la Gaussienne par une Laplace (0,1) mise à l'échelle.	78
A.1	Fonction de densité de la distribution triangulaire . . . . .	142
A.2	The symmetric beta density with parameters $\alpha = \beta = 1/16, 1/2, 1,$ and 3. . . . .	148
A.3	The beta density with parameters $\beta = 2$ and $\alpha = 0.3, 1,$ and 4. . .	149
A.4	Fonction de répartition empirique . . . . .	157
A.5	Fonction de répartition empirique lissée . . . . .	158
A.6	Fonction de répartition empirique prolongée . . . . .	158

# Liste des tableaux

4.1	générateurs avec $m = 2^e$ . . . . .	46
6.1	Bornes $(n - 1)/x_1$ et $(n - 1)/x_2$ d'un intervalle de confiance sur $\sigma^2/S_n^2$ . . . . .	87
6.2	Heuristique de Welch . . . . .	104
7.1	Nombre d'agents $n_j$ et taux d'arrivée $\lambda_j$ (par heure) pour 13 périodes d'une heure dans le centre d'appel. . . . .	114
7.2	Centre d'appel : tests sur les variables aléatoires communes . . . . .	120
A.1	Distribution multinomiale . . . . .	140
A.2	Distribution triangulaire . . . . .	143
A.3	Distribution de Erlang . . . . .	145
A.4	Distribution exponentielle . . . . .	145
A.5	Distribution de Cauchy . . . . .	146
A.6	Loi normale multivariée . . . . .	153
C.1	Exemples de files d'attente . . . . .	173



# Chapitre 1

## Introduction

Le présent document se veut un support partiel au cours IFT-3245 donné au Département d'Informatique et de Recherche Opérationnelle de l'Université de Montréal. Il est fourni en l'état, sans garantie sur l'absence d'erreurs. Celles-ci peuvent être signalées par courriel à l'adresse `bastin@iro.umontreal.ca`. L'exposé est basé en grande partie sur l'ouvrage de Law [8], ainsi que les notes du cours IFT-6561, *Simulation : aspects stochastiques*, écrites par Pierre L'Ecuyer. Les autres références seront mentionnées en temps utiles et sont reprises dans la bibliographie en fin d'ouvrage.

### 1.1 Concepts de base

Dans le cadre de la simulation, le but principal est de reproduire, ou plus précisément de simuler, le fonctionnement de divers processus, par exemple physiques, chimiques, . . . , d'industries, de chaînes de production, etc. Lorsque nous parlerons de simulations, nous emploierons les termes décrits ci-dessous.

**Système** Le processus d'intérêt sera généralement un système, qui peut être défini, en suivant Schmidt et Taylor [15] comme une **collection d'entités**, par exemple des hommes et/ou des machines, qui **agissent et interagissent** afin d'accomplir une certaine fin logique. L'**état d'un système** est la collection de variables nécessaires pour décrire un système à un instant particulier.

**Modèle** Un modèle, qui est une **description simplifiée** d'un système, dans le but d'évaluer (scientifiquement) sa performance ou l'effet de certaines décisions. Un modèle peut être physique (version à l'échelle du système : maquette, . . . )

ou mathématique (abstraction – simplifiée – sous forme mathématique).

**Simulation** La simulation, qui consiste à **faire évoluer le modèle d'un système** en fournissant les entrées appropriées, puis à **observer et analyser les résultats**.

Nous pouvons distinguer deux types principaux de systèmes : **discrets et continus**. Un système discret est un système dans lequel les variables d'état changent instantanément à des points séparés du temps, tandis qu'un système continu est un système dans lequel les variables d'états changent continûment par rapport au temps. Généralement, un système n'est ni totalement discret, ni totalement continu, bien qu'en pratique un type de comportement prévaut habituellement.

### 1.1.1 Pourquoi simuler ?

La simulation permet d'expérimenter avec un système sans payer le véritable prix de nos erreurs. Ses avantages peuvent être résumés comme suit :

- la simulation est non destructrice, et les erreurs ne sont pas (trop) coûteuses ;
- le système considéré n'a même pas besoin d'exister ;
- nous pouvons répéter à volonté des expériences identiques ou similaires dans les mêmes conditions ;
- nous pouvons souvent simuler un système beaucoup plus rapidement que son évolution dans la réalité, comme par exemple l'évolution d'un biosystème ou la formation du système solaire ;
- nous pouvons simuler des modèles très complexes, plus réalistes que ceux que nous pourrions résoudre par des formules analytiques ou par les méthodes d'optimisation classique.
- l'animation graphique peut permettre de voir évoluer le modèle.

Elle n'est cependant pas sans inconvénients :

- le coût de mise en œuvre peut être significatif, en particulier la modélisation et la programmation peuvent demander beaucoup d'effort, de temps et d'argent ;
- les temps d'exécution (CPU) peuvent devenir excessifs ;
- la simulation ne fournit habituellement que des estimations ; l'optimisation est beaucoup plus difficile par simulation que via les outils habituels de programmation mathématique (tels que ceux étudiés dans le cours IFT1575 par exemple) ;
- l'analyse statistique des résultats n'est pas toujours simple.

La simulation est présente de plus en plus dans notre environnement, notamment en raison du développement des capacités informatiques à notre disposition. Un exemple bien connu d'application concerne les simulateurs de vols, soit à des fins

ludiques, soit à des fins de formation des pilotes, notamment pour les entraîner à faire face à des situations d'urgence. Par conséquent, les simulations peuvent être de nature interactive, ce que nous ne considérerons pas dans le présent cours.



FIGURE 1.1 – Capture d'écran du simulateur open source FlightGear Flight Simulator

### 1.1.2 Modèles

Construire un modèle dépend du système que nous souhaitons étudier, et les différentes possibilités qui s'offrent sont résumées dans la Figure 1.2. Nous nous intéresserons ici en particulier aux modèles mathématiques, qui peuvent eux-mêmes être de différentes natures, comme décrit ci-dessous.

**Modèle analytique** Un modèle analytique est un modèle qui peut être décrit au moyen d'une formule mathématique. Une telle formule est généralement difficile à mettre au point, mais elle est facile à utiliser. Un exemple bien connu en théorie des files d'attente est la formule de Little (décrite à la Section C). Un modèle de simulation sera habituellement plus détaillé et réaliste qu'un modèle analytique. Par contre, une formule analytique peut donner

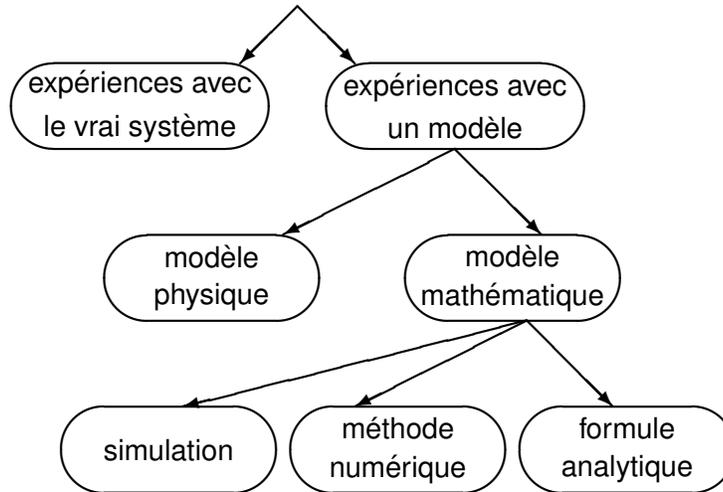


FIGURE 1.2 – Types de modèles

une meilleure intuition des principales propriétés du système et sera moins coûteuse à utiliser.

**Exemple 1.** *L'hypothèse première en modèles de flot de trafic est que la vitesse est une fonction décroissante de la densité. Comme la densité croît, l'espace entre les véhicules diminue, et les conducteurs réagissent en ralentissant. Un des modèles les plus populaires est la formule de Greenshields :*

$$u(K) = u_f - \frac{u_f}{K_j} K,$$

où  $u_f$  est la vitesse à flot libre (qui est observé pour une densité nulle) et  $K_j$  est la densité de congestion (aucun véhicule ne bouge plus).

**Méthodes numériques** Il s'agit de méthodes itératives et/ou approximatives. Nous pourrons ainsi recourir par exemple à la programmation dynamique, aux systèmes linéaires pour des chaînes de Markov imposantes, aux méthodes éléments finis pour des équations différentielles.

**Méthodes stochastiques** Ces approches impliquent la prise en compte de l'incertitude, et s'opposent par définition aux modèles déterministes. En conséquence, elles ont pour outil de base les probabilités et la statistique.

D'autres distinctions peuvent être opérées. Ainsi, un modèle **statique** est la représentation d'un système à un moment donné. Des exemples de modèles de simulation statiques sont certains modèles Monte Carlo. A l'opposé, un modèle de simulation **dynamique** représente un système qui évolue au cours du temps. A l'instar des systèmes, un modèle pourra également être discret ou continu. La décision d'utiliser d'utiliser un modèle discret ou continu dépend avant tout des objectifs de l'étude. Ainsi, un modèle discret pourra être utilisé pour représenter un modèle continu et vice-versa.

Nous nous concentrerons ici sur les **modèles de simulation à événements discrets**, qui sont des modèles discrets, dynamiques et stochastiques. Ces modèles seront étudiés avec plus de détail au Chapitre 3. L'organisation du syllabus est décrite au sein de la Figure 1.3.

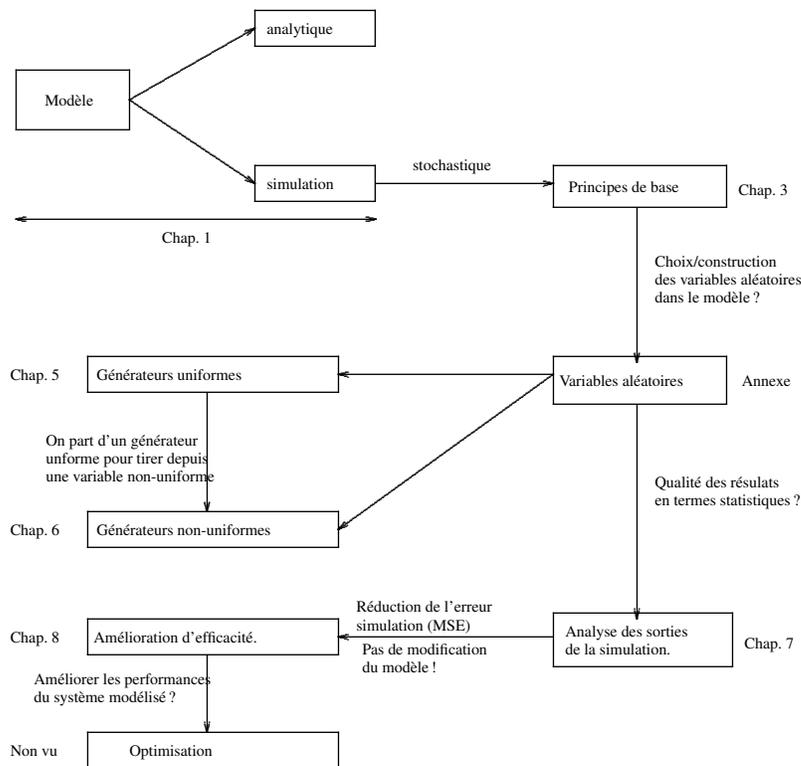


FIGURE 1.3 – Organisation des notes de cours

## 1.2 Logiciels de simulation

Le choix d'un logiciel de simulation est une étape importante pour mener à bien l'étude d'un modèle. Si le logiciel choisi n'est pas assez flexible ou trop difficile à utiliser, le projet de simulation peut produire des résultats erronés, voire tout simplement ne pas être conduit à terme. Une analyse du choix du logiciel de simulation est conduite dans Law [8], Chapitre 3. Deux tendances s'opposent en simulation, à savoir l'utilisation d'un paquetage de simulation et celle d'un langage de programmation classique. Chaque approche a ses avantages et ses inconvénients. Les paquetages de simulation présentent les caractéristiques suivantes :

- ils fournissent la plupart des caractéristiques nécessaires pour construire un modèle de simulation ;
- ils possèdent un environnement naturel pour la simulation ;
- les modèles de simulation sont généralement plus faciles à écrire et à modifier ;
- ils intègrent généralement beaucoup de vérification d'erreurs, ce qui facilite la tâche de vérification (voir Chapitre 2) ;
- nous pouvons en outre distinguer les paquetages de simulation à but générique, et les paquetages de simulation orientés pour des applications particulières.

A l'inverse, un langage de programmation se caractérisera par les points suivants :

- beaucoup de modélisateurs connaissent déjà un langage de programmation ;
- un modèle de simulation en C, C++, Java, . . . peut être plus rapide à exécuter ;
- plus de flexibilité de programmation ;
- nous pouvons bénéficier de l'approche orienté-objet ;
- le logiciel de développement coûte moins cher, bien que le coût total du projet puisse être plus important étant donné que le temps de développement en particulier sera souvent plus long.

Historiquement, les langages de simulation offraient un meilleur niveau de flexibilité, mais étaient souvent difficile à utiliser. Les simulateurs orientés applications bénéficiaient quant à eux d'avantage d'interfaces graphiques, étaient plus facile d'usage, mais pouvaient manquer de souplesse pour certains problèmes. Ces distinctions tendent toutefois à disparaître comme les langages de simulations sont à présent souvent intégrés dans des environnements graphiques, et les logiciels de simulation gagnent quant à eux en flexibilité. La distinction majeure reste donc sur la classe des simulations que peut traiter un logiciel donné, variant entre une orientation vers des applications spécifiques, ou au contraire une approche plus générique, même si le logiciel peut disposer en outre de caractéristiques spécifiques pour certaines applications.

**Exemple 2** (Micro-simulation en transport). *Un domaine très riche en simulation*

est la micro-simulation de transport, qui a pour objectif général de mieux comprendre et améliorer les réseaux de transports, de favoriser la sécurité routière, etc. Dans ces simulateurs, les interactions individuelles sont prises en compte, et de nombreux modèles, tels que les modèles de poursuite et de changements de bande, reflètent ces dépendances. Il est difficile d'interpréter les résultats de telles simulations autrement qu'en termes macroscopiques (nombre de véhicules par heure sur une portion de route donnée par exemple). La complexité de tels modèles a conduit à la création de divers outils spécifiques, accompagnés de langages de simulations adaptés. La visualisation graphique est un support courant et populaire dans ceux-ci, avec parfois une exagération des détails, donnant une impression de jeu vidéo plus qu'outil professionnel. La Figure 1.4 est une capture d'écran du simulateur Quadstone Paramics (<http://www.paramics-online.com/>). L'applet Java Traffic Microsimulation, écrite par Martin Treiber et disponible à

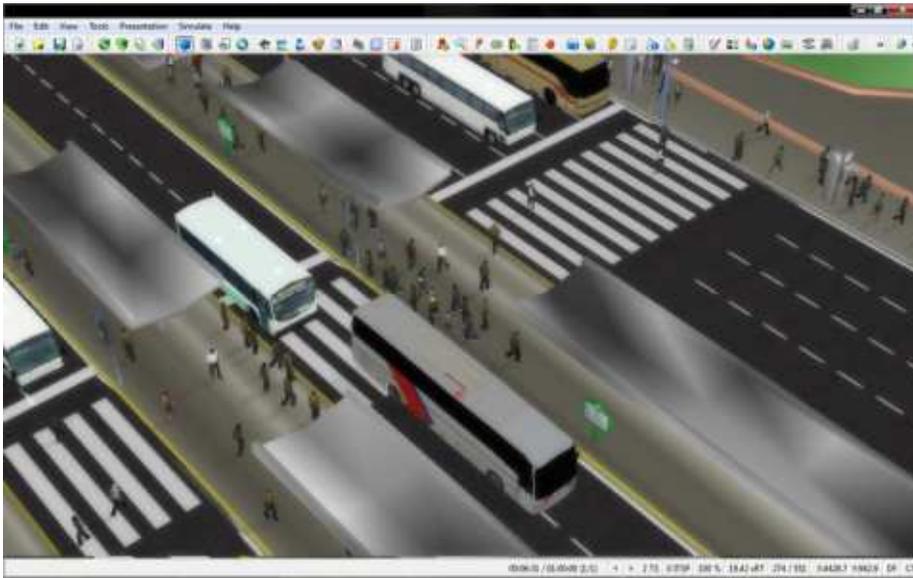


FIGURE 1.4 – Capture d'écran du micro-simulateur Paramics

l'adresse [www.traffic-simulation.de](http://www.traffic-simulation.de), illustre différents modèles de base de micro-simulation en transport.

Nous pourrions aussi occasionnellement être tenté d'utiliser des tableurs pour traiter des problèmes simples de simulation ; toutefois, les tableurs ont d'importantes limitations :

- seules des structures de données simples sont disponibles ;
- les algorithmes complexes sont difficiles à implanter ;
- les simulations par tableur peuvent requérir des temps d’exécution plus longs que les simulations construites dans un paquetage de simulation à événements discrets ;
- le stockage de données est limité.

Dans le cadre qui nous intéresse, nous utiliserons principalement la librairie open source SSJ (Stochastic Simulation in Java), qui peut être vu comme un langage de simulation générique, avec des outils plus spécifiques aux files d’attente, mais qui s’exprime dans le langage de programmation Java, et de ce fait, est à la frontière de la classification que nous avons opérée entre paquetages de simulation et langage de programmation. SSJ est développé à l’Université de Montréal, et téléchargeable à la page <http://www.iro.umontreal.ca/~simardr/ssj/>. SSJ contient différents paquetages offrant des outils pour générer des valeurs aléatoires uniformes et non uniformes, pour calculer différentes quantités liées à des lois de probabilité, effectuer des tests d’ajustement (“goodness-of-fit”), appliquer des méthodes de type quasi-Monte Carlo, et programmer des simulations à événements discrets, par événements ou par processus. La documentation officielle, ainsi qu’un recueil d’exemples, sont disponibles sur la page de référence de SSJ, aussi invitons-nous le lecteur à la visiter pour de plus amples informations. Nous utiliserons aussi occasionnellement d’autres outils, ordinairement écrits en C, en particulier la librairie ORATIO, disponible à l’adresse <http://amlet.slashbin.net>.

### 1.3 Applications

Nous illustrerons la simulation à événements discrets au travers de deux champs d’application : la simulation de trafic et les files d’attente.

### 1.4 Exercices

1. Partant de la Figure 1.2, déterminez à quel type de modèle correspond
  - (a) l’équation de vitesse  $v = at$ , où  $a$  est l’accélération et  $t$ , le temps ;
  - (b) un logiciel représentant graphiquement un processus de fabrication au sein d’une usine, en tenant compte de possibles incertitudes comme certaines pannes ;

- (c) un test aérodynamique en soufflerie du prototype d'un nouveau modèle de voiture ;
- (d) l'étude d'un processus de séparation au sein d'une colonne de distillation, composés de divers plateaux, en considérant deux flux opposés, lesquels voient leur composition changer à chaque plateaux. La composition des flux est calculée de plateau en plateau, en partant du bas de la colonne.



## Chapitre 2

# Modélisation : principes de base

*“All models are wrong, but some are useful.”*

George E. P. Box

### 2.1 Introduction et définitions

Une des difficultés majeures en simulation est d’essayer de déterminer si le modèle de simulation est une description précise du système réel en cours d’étude, c’est-à-dire si le modèle est valide. Avant d’aborder plus en détails comment valider un modèle, il est utile d’introduire quelques concepts supplémentaires, tels que présentés dans Law [8], Chapitre 5.

**Vérification** Le programme implante-t-il le modèle correctement ? Autrement dit, il s’agit de s’assurer que le programme informatique est correctement écrit, et répond aux hypothèses de travail fournies.

**Validation** Le modèle est-t-il assez précis pour l’application visée ? Est-ce une représentation précise du système, par rapport aux objectifs de l’étude ?

**Crédibilité** Les utilisateurs potentiels ont-ils suffisamment confiance dans le modèle pour l’utiliser ? Un modèle crédible n’est pas nécessairement valide, et vice versa. La crédibilité n’assure pas non plus que le modèle sera effectivement utilisé dans le processus de prise de décision.

La validité est relative, jamais absolue ; elle dépend de l’objectif du modèle. Conceptuellement, un modèle de simulation est valide s’il peut être utilisé pour prendre des décisions au sujet du système de façon similaire à celles qui seraient

prises s'il était possible et économiquement efficace d'expérimenter avec le système lui-même. La première question à se poser est dès lors de déterminer à qui servira ce modèle ? Le niveau de détail approprié en dépend. Un bon principe est de minimiser la complexité du modèle sous la contrainte qu'il donne une assez bonne approximation pour les besoins préalablement définis. Le principe général est donc de regarder la forêt avant les arbres, et de commencer par un modèle plus simple, approximatif, moins coûteux. Un modèle simple et réaliste (lorsqu'il existe) apporte beaucoup : il nous fait comprendre ce qui est le plus important. Il convient aussi d'être parcimonieux dans les hypothèses et les paramètres, et de ne raffiner le modèle que si nécessaire (et profitable). Il est de plus inutile, voire contre-productif, de construire un modèle trop fin compte tenu des données disponibles. Une bonne connaissance du système à modéliser et du problème à résoudre est primordiale ; il est souvent nécessaire d'interagir avec les spécialistes du domaine. Le modèle et le programme doivent être flexibles, faciles à modifier ; le principe de modularité doit primer autant que possible. La modélisation/simulation est un processus itératif, qui demeure en grande partie un art ; il n'y a pas de "recette" universelle.

Si le modèle sert à répondre à plusieurs questions, la validité se pose pour chacune d'elles. De plus, chaque fois qu'une nouvelle application est considérée pour le modèle de simulation, sa validité doit être reconsidérée. Le nouvel objectif peut-être substantiellement différent de l'objectif initial, et le temps passé depuis peut avoir invalidé certains paramètres du modèle. La validation coûte cependant cher, et ce d'autant plus que nous souhaitons le faire. De même, un modèle plus réaliste sera plus coûteux, aussi doit-on s'interroger s'il justifie l'investissement ? Nous devons faire un compromis entre d'une part les coûts de modélisation et de validation, et d'autre part les coûts associés aux conséquences d'un modèle non valide (ou pas suffisamment réaliste).

## 2.2 Techniques de validation

### Validation informelle, à vue

- Est-ce que tout semble raisonnable ? Fait-on les bonnes hypothèses ?
- Montrer aux experts ou aux clients. Expérience, intuition. Test de Turing.
- Animation graphique.
- Essayez des situations extrêmes.

### Tests statistiques des hypothèses

- Testez les approximations, les lois de probabilité, les hypothèses, etc.
- Comparaisons graphiques, intervalles de confiance, tests d'hypothèses.

- Attention : les tests formels ne sont pas nécessairement appropriés. On sait d'avance qu'il y aura des différences entre la sortie de la simulation et la réalité. Peut-on déterminer si cette différence est acceptable ?
- Analyse de sensibilité. (comparaison de deux situations proches).

#### **Validation Opérationnelle (comportement du modèle)**

- Comparer avec un système existant. *Calibration* (déterminer les valeurs des paramètres du modèle).
- Valider ensuite avec un ensemble *indépendant* de données. (A-t-on assez de données ?)
- Hypothèse à tester : le modèle est valide pour le niveau de précision acceptable, sous les conditions expérimentales.
- Est-ce que le modèle prédit bien l'avenir à *Validation prospective*. (Quid avec de nouvelles données ?)
- Tests de Turing. Crédibilité.

Pour des discussions sur les problèmes de validation et de modélisation, le lecteur pourra aussi consulter Cao [3].

## **2.3 Qualité des données**

Il faut de l'information *claire et fiable* sur le système, des données *pertinentes, correctes*, et en *quantité* suffisante. On a rarement les données que l'on veut !

*Peu de données :*

- échantillon trop petit.
- seulement des *résumés* (moyenne, variance, etc.).
- pas assez *précis* (e.g., durées des appels téléphoniques...).
- information *indirecte*, par ex. des entrevues.

Des données venant d'un *système proche*, mais différent.

- Trop d'*agrégation* (e.g., mensuel au lieu de quotidien).
- Mauvaise *période* (e.g., on a il y a 3 ans, mais on voudrait l'an prochain).
- Mauvais *endroit* (e.g., Toronto au lieu de Montréal).
- Distributions *censurées* (e.g., on veut les demandes, on a les ventes).

Le processus de validation doit inclure la validation des données. Si les données sont incertaines, il faut faire de l'analyse de sensibilité, pas seulement par rapport aux paramètres, mais aussi par rapport aux types de lois.

### 2.3.1 Hypothèses simplificatrices

Il est souhaitable de garder le modèle aussi simple que possible ; toutefois, il arrive fréquemment que des hypothèses simplificatrices soient avant tout le fruit de commodités afin de rendre le modèle techniquement plus facile à gérer. De telles simplifications peuvent toutefois nuire gravement à la qualité des résultats, et il conviendra de s'assurer de leur justesse. En particulier, divers types d'hypothèses sont habituellement opérées dans la modélisation des distributions d'entrées.

**Distributions simplifiées** Les distributions compliquées et inconnues du monde réel sont approximées par des distributions plus simples et plus élégantes, telles que l'exponentielle, la normale et l'uniforme. Souvent, certaines variables aléatoires seront remplacées par des constantes, par exemple leur espérance, ce qui peut modifier les valeurs calculées par le modèle.

**Exemple 3. Modèles linéaires généralisés** Dans un modèle linéaire généralisé, chaque réalisation de la variable dépendante  $Y$ , est supposé être générée à partir d'une distribution particulière de la famille exponentielle, qui inclue entre autres la normale, la binomiale, la Poisson, ... La moyenne,  $\mu$ , de la distribution dépend des variables indépendantes  $X$ , à travers la relation

$$E[\mathbf{Y}] = \mu = g^{-1}(X\beta),$$

où  $X\beta$  est le prédicteur linéaire, une combinaison linéaire des paramètres inconnus,  $\beta$ , et  $g$  est la fonction de liaison. Supposer que  $\mathbf{Y}$  suit une log-normale ou une gamma donne souvent des modèles assez similaires, néanmoins, il arrive que ces deux distributions, bien de forme similaire, conduisent à des résultats différents. Wiens [20] présente ainsi l'étude de la performance de la combinaison d'un vaccin et immunoglobuline anti-D à partir de données cliniques réelles. Supposer une log-normale conduit à la conclusion que la combinaison est bénéfique pour le patient, tandis que supposer une gamma indique qu'aucun gain n'est obtenu par rapport à l'utilisation du vaccin seul. Dans pareil cas, l'auteur suggère de compléter l'étude au moyen de méthodes visuelles et non-paramétriques.

**Aggrégation** Ce phénomène apparaît quand plusieurs types d'objets sont considérés comme formant une seule classe. Il peut s'agir de périodes de temps (e.g. des événements se produisant au cours de la même minute sont supposés avoir lieu simultanément), de personnes (e.g. aggrégation par classe d'âge et/ou de sexe), des objets, des ressources, etc.

**Indépendance** Il est usuel de supposer que certaines variables aléatoires dans le modèle sont indépendantes (au sens statistique du terme), malgré le fait

que ceci soit rarement vérifié dans la réalité. Par exemple, des temps de service successifs à un poste donné dans un système de file d'attente sont souvent supposés indépendants car modéliser la dépendance apparaît trop compliqué, ou il n'y a pas assez de données pour le faire. C'est probablement une source importante d'erreur évidente. Une des principales raisons pour les modélisateurs de supposer l'indépendance est que les logiciels commerciaux pour construire des modèles de simulation à événements discrets fournissent des outils facile d'usage presque exclusivement pour le cas de variables aléatoires i.i.d. Une raison plus sérieuse est la difficulté de modéliser proprement les dépendances.

**Stationnarité** Une modèle stationnaire est un modèle dont la logique, les distributions d'entrée et les paramètres ne changent pas avec le temps. Un tel modèle est plus facile à manipuler qu'une version non-stationnaire. Une telle hypothèse est typiquement raisonnable si le système d'intérêt est simulé sur une fraction relativement courte de sa durée de vie. Cependant beaucoup de modèles, en particulier ceux impliquant des humains, sont hautement non-stationnaires. Les résultats de simulations basées sur des modèles stationnaires, alors que les phénomènes sont non-stationnaires, peuvent être très suspects. Des hypothèses de stationnarité du modèle sont néanmoins généralement nécessaires afin d'obtenir des formules analytiques, ce qui explique également au moins en partie leur popularité. Pour la simulation, la stationnarité n'est pas nécessaire ; l'analyse des sorties peut même être plus facile pour des systèmes à horizon fini non-stationnaire que pour des modèles stationnaires en état stable.

### 2.3.2 Robustesse.

Une technique importante pour déterminer quels facteurs du modèle ont un impact significatif sur les mesures de performance souhaitées est l'analyse de sensibilité, laquelle consiste à étudier le comportement du modèle lorsque nous perturbons légèrement un facteur. Si un facteur particulier apparaît être important, il doit alors être modélisé avec soin. Les facteurs qui peuvent être investigués avec une analyse de sensibilité peuvent être

- la valeur d'un paramètre ;
- le choix d'une distribution ;
- l'entité se déplaçant à travers le système simulé ;
- le niveau de détail d'un sous-système ;
- les données les plus cruciales à obtenir (en utilisant en première étape un modèle grossier du système).

**Exemple 4** (File  $M/G/1$  à l'état stationnaire). Soit  $\lambda$ , le taux d'arrivée (loi exponentielle),  $\nu$  et  $\sigma^2$  la moyenne et la variance des durées de service, respectivement. Comme introduit dans la Section C, le facteur d'utilisation est donné par  $\rho = \lambda\nu$ . La formule de Pollaczek-Khintchine donne le nombre moyen dans le système :

$$L = \rho + \rho^2 \frac{(1 + \sigma^2/\nu^2)}{2(1 - \rho)}.$$

de sorte que le temps moyen vaut, en vertu de (C.2),

$$w = \frac{q}{\lambda} = \frac{\lambda(\sigma^2 + \nu^2)}{2(1 - \rho)}. \quad (2.1)$$

Ces quantités ne dépendent que de  $\lambda$ ,  $\nu$ , et  $\sigma$ . Mais si les interarrivées ne sont pas exponentielles, ou pour d'autres mesures de performance, ce n'est plus vrai. Quant à l'erreur introuite si nous approxime la file  $M/G/1$  par  $M/M/1$ , elle dépend de  $\sigma/\nu$  car pour une  $M/M/1$ ,  $\sigma = \nu$ .

## 2.4 Types de méthodes pour choisir les lois

### A. Paramétrique.

Choisir une loi standard compacte et estimer les paramètres.

Types de lois : exponentielle, normale, gamma, etc. ; estimer ; test d'ajustement.

Variables dépendantes : lois multivariées.

### B. Loi quasi-empirique ou non paramétrique (voir Annexe A).

Construire une variante de la fonction de répartition empirique et l'utiliser pour générer les variables. Méthodes d'estimation de densité.

### C. Simulation par retraçage.

Rejouer simplement l'histoire (le "log") d'un système réel, en changeant seulement les décisions et leurs conséquences : on tire au hasard dans l'historique. Peut être partiel.

Exemples : patients dans un hôpital, United Airlines.

L'inconvénient majeur de cette approche est qu'on ne produira jamais d'autres combinaisons que celles qui se trouvent dans les données, or il y a rarement assez de données pour réaliser toutes les simulations voulues. *Solution* : il faut modéliser les dépendances complexes ! Difficile...

Quelle méthode choisir ? Les recommandations générales suivantes peuvent être faites.

1. On privilégiera l'approche paramétrique s'il y a une certaine justification pour une loi particulière, mais il est alors important de bien comprendre les principes sous-jacents aux différentes lois ;
2. L'approche empirique se justifie en particulier si on a vraiment beaucoup de données, et à l'avantage par rapport à la méthode par retraçage est de pouvoir, pour des données continues, générer n'importe quelle valeur entre les points de données observées minimum et maximum ;
3. On préférera l'approche par retraçage s'il y a beaucoup de données et des dépendances trop difficiles à modéliser. Cette approche est aussi recommandée pour la validation de modèles, quand la sortie du modèle pour un système existant est comparée avec le sortie correspondante pour le système lui-même.

### 2.4.1 Approche paramétrique

Arguments *favorisant* l'approche paramétrique *A* :

- Elle est appropriée si on a des raisons “physiques” de choisir une loi particulière. Toutefois, même si nous sommes assez chanceux et disposons de ce type d'information, il est toujours conseillé d'utiliser les données observées pour fournir un support empirique à l'usage d'une distribution particulière.
- C'est une façon compacte de représenter les données.
- Les procédures disponibles pour générer les variables aléatoires sont habituellement plus souples, requérant moins de travail de programmation. Une distribution théorique est aussi plus facile à modifier.
- Les valeurs que l'on peut générer ne sont pas limitées à l'étendue de l'échantillon (important si on veut pouvoir générer des événements “extrêmes”). Cela “lisse”, i.e. régularise, les données, et peut fournir de l'information sur toute la distribution sous-jacente. Une fonction de distribution empirique peut avoir certaines “irrégularités”, en particulier si seulement un petit nombre de valeurs de données sont disponibles. De plus, avec les fonctions empiriques standards, il n'est pas possible de générer au cours de la simulation des valeurs en dehors du domaine des valeurs observées. Or, nombre de mesures de performance pour des systèmes de simulation dépendent fortement de la probabilité de réalisation d'un événement “extrême”, par exemple un très grand temps de service. Avec une distribution théorique calibrée, il est possible de générer des valeurs en dehors du domaine des données observées.

Arguments *contre A* :

- Il est souvent très difficile, parfois même impossible, de connaître le bon

type de loi. Parfois, les données ne s'ajustent à aucune des lois disponibles. C'est en particulier le cas si les données proviennent de deux populations hétérogènes, ou plus, ou si les données ont été trop arrondies, de sorte qu'il n'y a pas assez de valeurs distinctes dans l'échantillon pour permettre à n'importe quelle distribution théorique continue de fournir une bonne représentation.

- L'estimation des paramètres n'est pas toujours robuste.
- Lors de l'ajustement, il y a perte ou distorsion d'information.
- Pour certaines lois, la génération de v.a. est difficile ou lente.

L'approche paramétrique est recommandée seulement si l'ajustement est bon, et sur la base de justifications théoriques. Les paramètres peuvent être habituellement classés, sur base de leur interprétation physique ou théorique, comme appartenant à un des trois types de base ci-dessous, en particulier dans le cas de distributions continues.

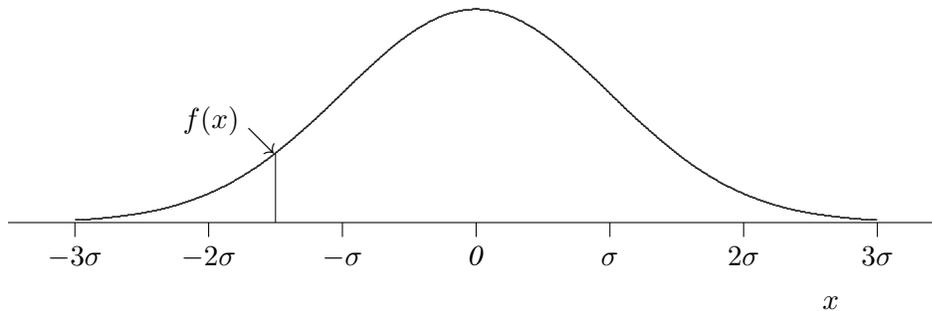
**Localisation** : spécifie l'origine sur l'axe des abscisses ; comme ce paramètre change, la distribution est simplement déplacée à gauche ou à droite, mais sans subir d'autres changements.

**Echelle** : détermine l'échelle sans changer la forme de la courbe ; modifier ce paramètre revient à étirer ou contracter la distribution, mais sans altérer sa forme de base.

**Forme** : modifie plus en profondeur la forme de la fonction de densité ou de masse, et les propriétés de la loi, à l'intérieur de la famille générale de distributions. Certaines distributions n'ont pas de paramètres de forme, tandis que d'autres peuvent en avoir deux.

Une *transformation affine* ( $Y = aX + b$ ) suffit pour contrôler la localisation et l'échelle.

**Exemple 5** (La loi normale.).  $X \sim N(\mu, \sigma^2)$  ssi  $Z = (X - \mu)/\sigma \sim N(0, 1)$ .  $X = \sigma Z + \mu$ .  $\mu$  est le paramètre de localisation, et  $\sigma$  le paramètre d'échelle. Il n'y a pas de paramètre de forme.



### 2.4.2 Estimation de paramètres

Supposons que nous disposons de  $n$  observations  $x_1, \dots, x_n$  venant d'une loi de densité  $f_\theta(x)$ , dépendant d'un certain  $\theta$  inconnu, et que nous souhaitons estimer  $\theta$ .  $x$  et  $\theta$  peuvent être des vecteurs. Un estimateur est une fonction numérique des données (voir Annexe 6.1). Il y a beaucoup de manière de spécifier la forme d'un estimateur pour un paramètre particulier d'une distribution donnée, et beaucoup de façons d'évaluer la qualité d'un estimateur. Toutefois, une des approches les plus courantes est le maximum de vraisemblance, pour les raisons suivantes :

1. les estimateurs de maximum de vraisemblance ont plusieurs propriétés désirables dont ne jouissent habituellement pas les autres méthodes ;
2. l'utilisation du maximum de vraisemblance se révèle importante dans la justification du test chi-carré de qualité d'ajustement ;
3. l'idée centrale de l'estimation par maximum de vraisemblance est intuitive.

Sur bases des observations  $x_1, \dots, x_n$ , nous pouvons définir la fonction *fonction de vraisemblance* de l'échantillon dans le cas discret comme

$$L(\theta) = p_\theta(x_1)p_\theta(x_2) \cdots p_\theta(x_n),$$

et dans le cas continu comme

$$L(\theta) = f_\theta(x_1)f_\theta(x_2) \cdots f_\theta(x_n).$$

Si nous considérons les variables aléatoires i.i.d.  $X_1, \dots, X_n$ , dont sont issues les réalisations  $x_1, \dots, x_n$ , nous pouvons voir dans le cas discret la fonction de vraisemblance comme la fonction de masse de probabilité jointe :

$$L(\theta) = p_\theta(X_1)p_\theta(X_2) \cdots p_\theta(X_n).$$

*Estimateur de vraisemblance maximale (EVM)* : c'est la valeur de  $\theta$  qui maximise  $L(\theta)$ . De plus, sous certaines conditions<sup>1</sup>,  $\sqrt{n}(\hat{\theta}_n - \theta) \Rightarrow N(0, n(\mathbf{I}(\theta))^{-1})$ , où  $\mathbf{I}(\theta)$  est la matrice d'information de Fisher, définie comme  $-E \left[ \frac{d^2}{d\theta^2} \ln L(\theta) \right]$ .

Autres méthodes : *ajustement des moments* ("moment matching"), *moindres carrés*,...

---

1. Le résultat ne tient pas en particulier si des contraintes sur  $\theta$  sont actives.

**Exemple 6.** Loi Weibull avec  $\delta = 0$  et  $\theta = (\alpha, \lambda)$ . On a

$$\begin{aligned} f(x) &= \alpha \lambda^\alpha x^{\alpha-1} e^{-(\lambda x)^\alpha} \quad \text{pour } x > 0, \\ L(\alpha, \lambda) &= \alpha^n \lambda^{n\alpha} (x_1 \cdots x_n)^{\alpha-1} e^{-\lambda^\alpha (x_1^\alpha + \cdots + x_n^\alpha)}, \\ \ln L(\alpha, \lambda) &= n \ln \alpha + n\alpha \ln \lambda + (\alpha - 1) \sum_{i=1}^n \ln x_i - \lambda^\alpha \sum_{i=1}^n x_i^\alpha. \end{aligned}$$

Examinons où la dérivée de  $\ln L$  vaut zéro, ce qui est plus facile que pour  $L$ . Nous obtenons

$$\begin{aligned} 0 = \frac{\partial \ln L(\alpha, \lambda)}{\partial \alpha} &= \frac{n}{\alpha} + n \ln \lambda + \sum_{i=1}^n \ln x_i - \sum_{i=1}^n \frac{d}{d\alpha} e^{\alpha \ln(\lambda x_i)} \\ &= \frac{n}{\alpha} + \sum_{i=1}^n (\ln \lambda + \ln x_i) - \sum_{i=1}^n \ln(\lambda x_i) e^{\alpha \ln(\lambda x_i)} \\ &= \frac{n}{\alpha} + \sum_{i=1}^n \ln(\lambda x_i) - \sum_{i=1}^n (\lambda x_i)^\alpha \ln(\lambda x_i), \\ 0 = \frac{\partial \ln L(\alpha, \lambda)}{\partial \lambda} &= \frac{\alpha n}{\lambda} - \alpha \lambda^{\alpha-1} \sum_{i=1}^n x_i^\alpha. \end{aligned}$$

La seconde équation permet d'écrire  $\lambda$  en fonction de  $\alpha$  :

$$\sum_{i=1}^n (\lambda x_i)^\alpha = n, \text{ ou } \lambda = \sqrt[\alpha]{\frac{n}{\sum_{i=1}^n x_i^\alpha}},$$

puis on remplace dans la première et on la résoud numériquement (vu qu'il n'y pas d'expression simple pour la dérivée par rapport à  $\alpha$ ).

Si on pose  $\alpha = 1$ , on obtient :  $\hat{\lambda}_n = 1/\bar{x}_n = n/\sum_{i=1}^n x_i^\alpha$ .

**Exemple 7** (Modèles de choix discrets). Les modèles de choix discrets fournissent une description de la manière dont les individus effectuent une sélection au sein d'une ensemble fini de possibilités. Soit  $I$  la taille de population et  $\mathcal{A}(i)$  l'ensemble des choix possibles pour l'individu  $i$ ,  $i = 1, \dots, I$ . Pour chaque individu  $i$ , chaque choix possible  $A_j$ ,  $j = 1, \dots, |\mathcal{A}(i)|$  possède, une certaine utilité, dépendant des caractéristiques de l'individu et l'attractivité relative de l'alternative, que nous supposons avoir la forme  $U_{ij} = V_{ij} + \epsilon_{ij}$ , où  $V_{ij} = V_{ij}(\beta_j, x_{ij})$  est une fonction d'un vecteur de paramètres  $\beta_j$  et de  $x_{ij}$ , les attributs observés de l'alternative  $A_j$ , tandis que  $\epsilon_{ij}$  est une terme aléatoire reflétant la partie non observée de l'utilité.

La théorie suppose que l'individu  $i$  sélectionne l'alternative qui maximise son utilité, i.e. la probabilité que l'individu  $i$  choisisse  $A_j$  est donnée par

$$P_{ij} = P[\epsilon_{il} \leq \epsilon_{ij} + (V_{ij} - V_{il}), \forall A_l \in \mathcal{A}(i)].$$

Une première question se pose sur le choix des termes aléatoires  $\epsilon_{ij}$ , qui influencent directement la forme des probabilités  $P_{ij}$ . Considérons le vecteur

$$\epsilon_i = (\epsilon_{i1}, \dots, \epsilon_{i|A_i}).$$

En supposant que  $\epsilon_i$  suit une normale multivariée  $N(O, \Sigma)$ , la probabilité de choix est donnée par

$$\begin{aligned} P_{ij} &= P[V_{ij} + \epsilon_{ij} \geq V_{il} + \epsilon_{il}, \forall l \neq i] \\ &= \int I(V_{ij} + \epsilon_{ij} \geq V_{il} + \epsilon_{il}, \forall l \neq i) \phi(\epsilon_i) d\epsilon_i, \end{aligned}$$

où  $I(\cdot)$  est la fonction indicatrice et  $\phi(\cdot)$  est la fonction de densité d'une normale multivariée. Il s'agit du modèle probit. En utilisant au contraire des distributions de Gumbel indépendantes, de facteur d'échelle 1.0, nous obtenons la formule logit :

$$L_{ij}(\beta) = \frac{e^{V_{ij}(\beta)}}{\sum_{l=1}^{|\mathcal{A}(i)|} e^{V_{il}(\beta)}}, \quad (2.2)$$

La formule (2.2) caractérise le modèle multinomial logit traditionnel (aussi appelé conditional logit). Bien que populaire, il souffre de la propriété d'indépendance des alternatives irrelevantes, en abrégé IIA (pour independence of irrelevant alternatives). Cette propriété énonce qu'ajouter une alternative ou modifier les caractéristiques d'une alternatives ne changera pas les rapports de probabilité entre les autres alternatives. Cette propriété n'est pas réaliste pour des applications présentant des alternatives similaires. Dans l'exemple célèbre des bus rouges/bleus (voir par exemple Ben Akiva et Lerman [1]), des navetteurs doivent initialement prendre une décision par rapport à deux modes de transport : la voiture ou le bus rouge. Supposons qu'un client choisisse de manière équiprobable, aussi le rapport des probabilités vaut 1. Supposons à présent que nous ajoutons l'alternative bus bleu. Il est raisonnable de présumer que les navetteurs ne seront pas affectés par la couleur des bus, aussi la probabilité de choisir la voiture restera 1/2, tandis les probabilités de chaque type de bus sera de 1/4. La propriété IIA implique que ce n'est pas le cas, mais que les trois alternatives ont chacune une probabilité 1/3.

Dans les modèles logits multinomiaux mélangés (MLMM), les vecteurs  $V_i$ ,  $i = 1, \dots, I$ , contiennent eux-mêmes des éléments aléatoires, et nous écrivons

$V_{ij} = g(\beta, x_{ij}, \xi_{ij})$ . Nous basant sur la formulation en coefficients aléatoires (voir par exemple Revelt and Train [14]), les termes aléatoires supplémentaires sont exploités pour introduire de l'hétérogénéité à travers les preneurs de décisions, de sorte que  $\beta$  devient lui-même un vecteur aléatoire. Nous supposons dès lors que  $\beta$  provient lui-même d'un vecteur aléatoire  $\omega$  et d'un vecteur de paramètres  $\theta$ , ce que nous exprimons comme  $\beta = \beta(\omega, \theta)$ . La probabilité de choix est alors donnée par

$$P_{ij}(\theta) = E_P [L_{ij}(\omega, \theta)] = \int L_{ij}(\omega, \theta) P(d\omega) = \int L_{ij}(\omega, \theta) f(\omega) d\omega,$$

où  $P$  est la mesure de probabilité associée à  $\omega$  et  $f(\cdot)$  est sa fonction de densité. Le vecteur de paramètres  $\theta$  est généralement estimé en maximisant la fonction de vraisemblance logarithmique, c'est-à-dire en résolvant le programme

$$\max_{\theta} LL(\theta) = \max_{\theta} \frac{1}{I} \sum_{i=1}^I \ln P_{ij_i}(\theta), \quad (2.3)$$

où  $j_i$  est le choix exprimé par l'individu  $i$ . L'approximation par échantillonnage moyen de (2.3) est :

$$\max_{\theta} SLL^R(\theta) = \max_{\theta} \frac{1}{I} \sum_{i=1}^I \ln SP_{ij_i}^R(\theta), \text{ avec } SP_{ij_i}^R(\theta) = \frac{1}{R} \sum_{r_i=1}^R L_{ij_i}(\omega_{r_i}, \theta),$$

où  $R$  est le nombre de tirs aléatoires  $\omega_{r_i}$ , provenant de la distribution de  $\omega$ .

Pour plus d'informations sur les modèles de choix discrets, le lecteur pourra notamment consulter Train [17].

## 2.5 Tests d'Ajustement

Après avoir déterminé une ou plusieurs distributions de probabilité qui pourraient s'ajuster aux données observées, il reste à vérifier si ces distributions représentent adéquatement la distribution sous-jacente pour nos données, et si plusieurs de ces distributions sont représentatives, nous devons aussi déterminer quelle distribution fournit le meilleur ajustement, pour les besoins du modèle<sup>2</sup>.

---

2. Pour plus de détails, le lecteur est renvoyé à Law [8], Section 6.6

### 2.5.1 Procédures heuristiques

#### Graphes histogramme-densité et comparaisons de fréquences

Pour un jeu de données continues, un histogramme est essentiellement un estimateur graphique du graphe de la fonction de densité correspondant à la distribution de nos données  $x_1, \dots, x_n$ , obtenues à partir des variables i.i.d.  $X_1, \dots, X_n$ , de densité  $f$ . Pour construire un histogramme, nous devons séparer le domaine des valeurs couvertes par nos données en  $k$  intervalles adjacents disjoints  $[b_0, b_1)$ ,  $[b_1, b_2)$ , ...,  $[b_{k-1}, b_k)$ . Tous les intervalles devraient avoir la même taille  $\Delta b = b_j - b_{j-1}$ ,  $j = 1, \dots, k$ , ce qui peut impliquer le rejet de quelques valeurs extrêmes. Pour  $j = 1, \dots, k$ , soit  $h_j$  la proportion des observations  $x_i$ ,  $i = 1, \dots, n$ , qui se trouvent dans le  $j^e$  intervalle  $[b_{j-1}, b_j)$ . Définissons la fonction

$$h(x) = \begin{cases} 0 & \text{si } x < b_0 \\ h_j & \text{si } b_{j-1} \leq x < b_j, \quad j = 1, \dots, k, \\ 0 & \text{si } b_k \leq x. \end{cases}$$

que nous traçons comme fonction de  $x$ . Soit  $X$  une variable de densité  $f$ . Pour n'importe quel  $j$ ,  $j = 1, \dots, k$ , nous avons en vertu du théorème de la valeur moyenne

$$P[b_{j-1} \leq X < b_j] = \int_{b_{j-1}}^{b_j} f(x) dx = \Delta b f(y),$$

pour un certain  $y \in (b_{j-1}, b_j)$ . D'autre part, la probabilité que  $X$  tombe dans le  $j^e$  intervalle peut être approximée par  $h(y)$ .

Les histogrammes sont applicables à n'importe quelle distribution continue, et fournissent un aperçu graphique des données directement interprétable. Toutefois, ils présentent certaines difficultés, en particulier il n'existe pas de guide absolu pour choisir le nombre d'intervalles  $k$ .

La fonction de masse de probabilité correspondant à un jeu de données discrètes peut aussi être estimée au moyen d'un histogramme. Pour chaque valeur possible  $a_j$ , soit  $h_j$  la proportion des  $x_i$  égaux à  $a_j$ . L'histogramme est obtenu en traçant des barres verticales  $h_j$  versus  $a_j$ . Dans le cas discret,  $h_j$  est un estimateur non-biaisé de  $p(a_j)$ , où  $p(\cdot)$  est la véritable fonction de masse des données. Nous n'avons pas besoin non plus de prendre des décisions subjectives sur la largeur des intervalles et leur placement.

Pour des données continues, un graphe histogramme-densité peut être réalisé en dessinant  $\Delta b \hat{f}(x)$  sur l'histogramme  $h(x)$  et en recherchant les similarités.

### Graphes de probabilité

*Graphique P-P* : points  $((i-0.5)/n, F(x_{(i)})) = (\hat{F}_n(x_{(i)}) - 0.5/n, F(x_{(i)}))$  ;  
L'idée est de dessiner les valeurs de  $\hat{F}_n(x)$  contre ceux de  $F(x)$ . On devrait avoir approximativement une droite de pente de 45?.

*Graphique Q-Q* : points  $(F^{-1}((i-0.5)/n), x_{(i)})$ .

On applique  $F^{-1}$  à chaque coordonnée des points du P-P plot.

Le P-P plot détecte les différences davantage au centre, le Q-Q plot davantage dans les queues.

Chi-deux, Kolmogorov-Smirnov, Anderson Darling.

*Kolmogorov-Smirnov* :

$$\begin{aligned} D_n &= \sup_{-\infty < x < \infty} |\hat{F}_n(x) - F(x)| \\ &= \max \left( \max_{1 \leq i \leq n} [i/n - u_{(i)}], \max_{1 \leq i \leq n} [u_{(i)} - (i-1)/n] \right). \end{aligned}$$

On rejette  $\mathcal{H}_0$  : "la bonne loi est  $F$ " si  $D_n$  est trop grand.

## 2.6 Exercices

1. Il est courant en sécurité routière de suggérer le maintien d'un écart de 2 secondes par rapport au véhicule qui précède. En considérant une distribution log-normale pour le temps de réaction, de paramètres  $\mu = 0.14$  et  $\sigma = 0.44$ , calculez à l'aide de SSJ le temps de réaction moyen, ainsi que les quantiles 0.90, 0.95. Sur base de ces résultats, considérez-vous la suggestion appropriée ? Considérons à présent l'utilisation d'une distribution normale, de moyenne 1.25 et d'écart-type 0.46. Répétez le calcul des quantiles 0.90 et 0.95. Quelle est la probabilité de générer un temps de réaction négatif ? Tronquez la distribution en 0 pour ne produire que des temps positifs. Que deviennent la moyenne, l'écart-type, les quantiles 0.9 et 0.95 ?

## Chapitre 3

# Simulation à événements discrets

### 3.1 Approche par événements

La simulation à événements discrets concerne la modélisation d'un système qui évolue au cours du temps, à l'aide d'une représentation dans laquelle des variables d'états peuvent seulement changer en un nombre dénombrable de points dans le temps. Plus précisément, des événements, c'est-à-dire une occurrence instantanée qui peut changer l'état du système,  $e_0, e_1, e_2, \dots$  surviennent aux instants  $0 = t_0 \leq t_1 \leq t_2 \leq \dots$ . Notons  $\mathcal{S}_i$  l'état du système immédiatement après  $e_i$ , et on définit le **temps de la simulation** comme étant la valeur courante de  $t_i$ .  $(t_i, \mathcal{S}_i)$  doit contenir assez d'information pour poursuivre la simulation (sauf les valeurs des variables aléatoires générées lors des événements  $e_j$  pour  $j > i$ ). Nous pouvons représenter graphiquement la simulation à événements discrets comme sur la figure 3.1, et résumer les composantes et l'organisation générale d'un modèle de simulation à événements discrets comme suit.

**Programme principal** Toute simulation commence par une phase d'initialisation qui définit l'état du système au temps 0. Autrement dit, nous définissons le temps courant  $(t_0, \mathcal{S}_0)$ . Afin de gérer correctement les différents ingrédients qui composent le paquetage de simulation, nous disposerons en général d'une routine d'initialisation qu'il conviendra d'appeler. Cette routine permettra en particulier de démarrer l'horloge de simulation, et de créer différentes variables internes, comme la liste d'événements (différentes implantations possibles).

**Horloge de simulation** L'horloge de simulation représente l'écoulement du temps

au sein de la simulation ; elle fera l'objet d'un traitement plus complet au sein de la Section 3.2.

**Liste d'événements** Chaque événement devra disposer de sa propre routine, servant à mettre à jour l'état du système. Le paquetage de simulation devra en outre disposer d'une routine de synchronisation afin de traiter le fait que l'occurrence d'un événement peut influencer les événements futurs.

**Exemple 8.** *La classe `Event` du paquetage `SSJ` contient entre autres les méthodes suivantes :*

- *actions, qui doit être définie par l'utilisateur, et qui décrit comment l'état du système est modifié par l'événement ;*
- *schedule, qui donne l'intervalle de temps après lequel l'événement surviendra, à partir de l'instant présent de la simulation ; cet instant peut être modifié à l'aide de la méthode `reschedule` et consulté avec la méthode `time`, qui donne la valeur de l'horloge de simulation au moment où est programmé l'événement.*
- *cancel permet de supprimer cet événement de la liste d'événements.*

**Routines de génération de nombres aléatoires** Comme nous considérons des simulations stochastiques, il convient de pouvoir traiter adéquatement les variables aléatoires qui interviennent dans le modèle.

### Compteurs statistiques

### Générateur de rapport

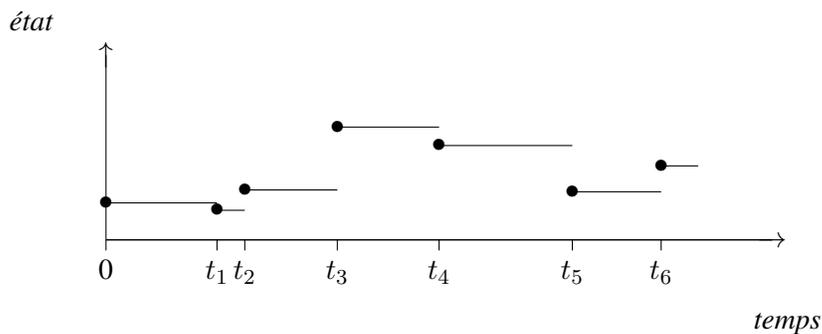


FIGURE 3.1 – Représentation schématisée d'une simulation à événements discrets

## 3.2 Le temps de simulation

En raison de la nature dynamique des modèles de simulation à événements discrets, nous devons garder trace de la valeur du temps de simulation au cours de la progression de celle-ci. Pour ce faire, nous conserverons dans une variable du modèle de simulation la valeur courante du temps de simulation. Cette variable est appelée **horloge de simulation**. Typiquement, il n'y a pas de relation entre le temps simulé et le temps réel d'exécution de la simulation. Historiquement, deux approches principales ont été suggérées pour avancer l'horloge de simulation : avancée au temps du prochain événement et avancée par incréments fixes de temps. La première approche est la plus utilisée, comme elle permet de se concentrer sur les changements d'état du système, tandis que la seconde peut être vue comme un cas particulier de la précédente.

Dans une approche par événements, l'horloge de simulation est initialisée à zéro, et les instants d'occurrence des événements futurs sont déterminés. L'horloge de simulation est alors avancée à l'instant d'occurrence de l'événement le plus imminent. Ce événement est traité et détruit, l'état du système est mis à jour en conséquence, de même que notre connaissance des instants d'occurrence des événements futurs. L'horloge de simulation est alors avancée jusqu'au prochain événement le plus imminent, l'état du système est mis à jour, les instants des futurs événements sont déterminés, ... Le processus d'avancement de l'horloge de simulation est poursuivi jusqu'à finalement atteindre certaines conditions d'arrêt, spécifiées au préalable. Vu que tous les changements d'état ont lieu seulement lors de l'occurrence d'événements, les périodes d'inactivité sont ignorées, en avançant l'horloge de simulation d'instant d'événement en instant d'événement. Ces sauts de l'horloge de simulation sont généralement variable en taille. La Figure 3.2 illustre le principe général d'une simulation à événements discrets.

**Exemple 9** (File d'attente à un serveur). *Considérons une file  $GI/G/1$ , comprenant donc un seul serveur, devant lequel les clients arrivent un à un et sont servis un à la fois, en ordre FIFO. De plus,*

- $S_i$  est la durée de service du client  $i$ ;
- $A_i$  est la durée entre les arrivées des clients  $i$  et  $i + 1$ .

*Les  $S_i$  et les  $A_i$  sont mutuellement indépendantes, et de fonctions de répartition  $G$  et  $F$ , respectivement. Le premier client arrive au temps  $A_0$  dans un système vide. Nous souhaitons simuler ce système pour une durée  $T$  et calculer l'attente moyenne par client et la longueur moyenne de la file d'attente. Les types d'événements sont*

1. arrivée;

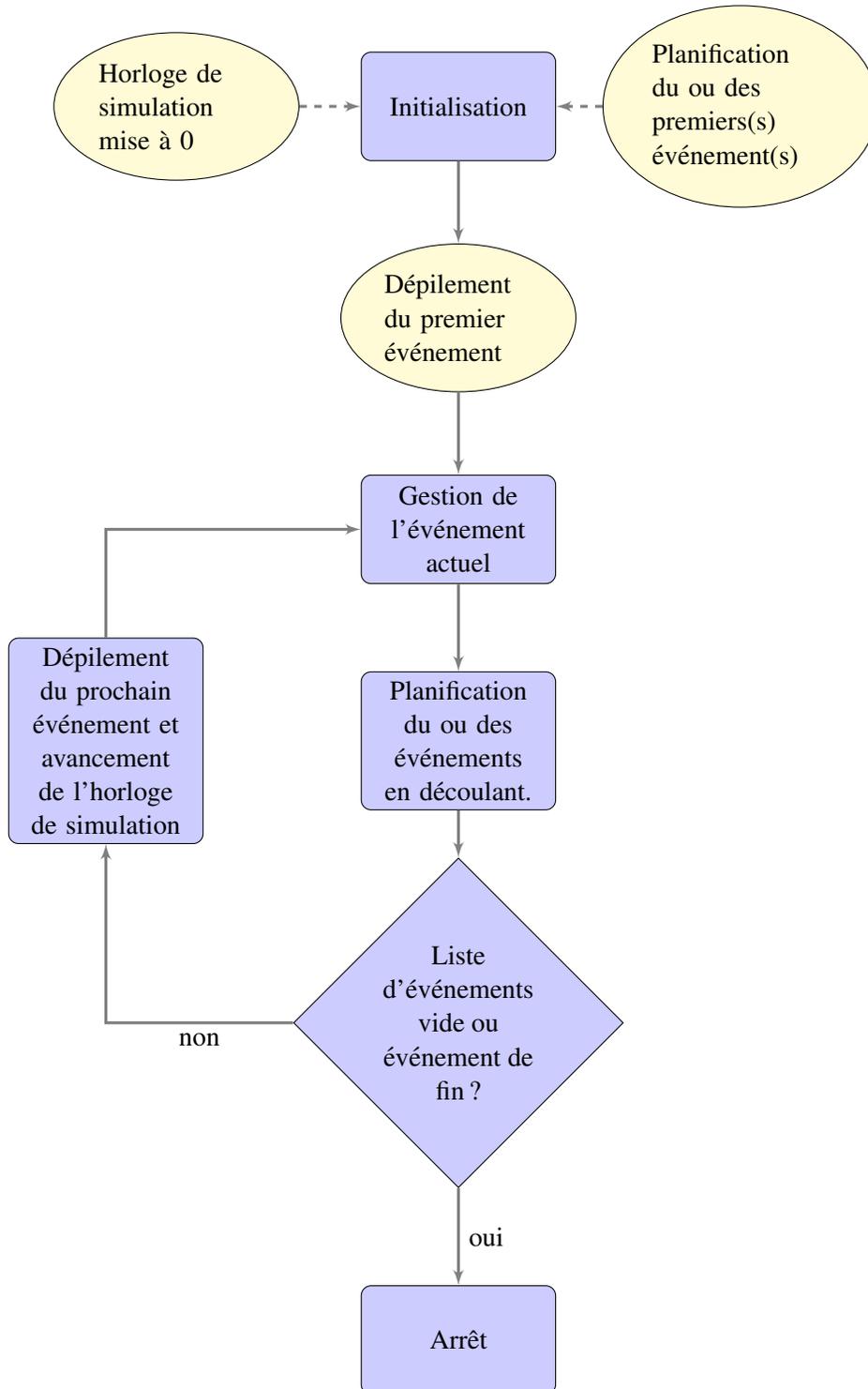


FIGURE 3.2 – Schéma d'une simulation à événements discrets.

2. départ ;
3. fin de la simulation.

Les variables aléatoires (indépendantes) à générer sont  $A_1, A_2, A_3, \dots$  et  $S_1, S_2, S_3, \dots$

Nous reprendrons les notations suivantes, comme introduites dans l'Annexe C :

- $W_i$ , le temps d'attente du client  $i$  ;
- $Q(t)$ , la longueur de la file d'attente au temps  $t$  ;
- $N_c(t)$ , le nombre de clients ayant débuté leur service durant l'intervalle  $[0, t]$ .

Supposons que nous voulons calculer, pour l'intervalle  $[0, T]$ , l'attente moyenne par client,

$$\overline{W}_{N_c(T)} = \frac{1}{N_c(T)} \sum_{i=1}^{N_c(T)} W_i;$$

et la longueur moyenne de la file d'attente :

$$\overline{Q}_T = \frac{1}{T} \int_0^T Q(t) dt.$$

Cette dernière intégrale est facile à calculer par simulation. Pour chaque client, nous créons un objet contenant l'instant d'arrivée et la durée de service. Les variables d'état reprennent la liste des clients en attente et la liste des clients en service. De plus, nous aurons comme dans toute simulation à événements discrets, l'horloge de la simulation, des compteurs statistiques (au besoin), une liste des événements futurs prévus, par ordre chronologique, et une procédure pour chaque type d'événement :

**Arrivée** Générer  $A$  selon  $F$  et prévoir une autre arrivée dans  $A$  unités de temps ;

Créer le nouveau client et noter son instant d'arrivée ;

Générer sa durée de service  $S$  selon  $G$  ;

**Si** (serveur est occupé) **alors**

Insérer ce nouveau client dans la liste des clients en attente ;

**sinon**

Insérer le client dans la liste des clients en service ;

Prévoir son départ dans  $S$  unités de temps ;

Mise à jour des statistiques voulues.

**Départ** Enlever le client de la liste des clients en service ;

**Si** la file d'attente n'est pas vide **alors**

Enlever le premier client de la liste d'attente ;

L'insérer dans la liste des clients en service ;

Récupérer son  $S$  et prévoir son départ dans  $S$  unités de temps ;

Mise à jour des statistiques voulues.

**Fin-de-la-Simulation** *Imprimer un rapport et terminer le programme.*

*Pour démarrer la simulation, il convient d'initialiser les variables et les compteurs, de prévoir la "Fin-de-la-Simulation" au temps  $T$ , de générer  $A$  selon  $F$  et de prévoir la première "Arrivée" dans  $A$  heures. Ceci fait, la simulation peut être lancée. L'exécution de la simulation consiste simplement à répéter la boucle suivante :*

**Répéter :** *exécuter le prochain événement dans la liste d'événements*

**jusqu'à :** *la liste des événements prévus est vide,*

*ou bien un événement stoppe la simulation.*

*A titre d'illustration, le programme suivant, basé sur SSJ, permet de simuler une file  $M/M/1$  :*

```
public class QueueEv {

    RandomVariateGen genArr, genServ;
    LinkedList waitList = new LinkedList ();
    LinkedList servList = new LinkedList ();
    Tally custWaits = new Tally ("Waiting times");
    Accumulate totWait = new Accumulate ("Size of queue");

    class Customer { double arrivTime, servTime; }

    public QueueEv (double lambda, double mu) {
        genArr = new RandomVariateGen (new RandMrg(),
                                       new ExponentialDist (lambda));
        genServ = new RandomVariateGen (new RandMrg(),
                                       new ExponentialDist (mu));
    }

    public void simulateOneRun (double timeHorizon) {
        Sim.init ();
        new EndOfSimulation ().schedule (timeHorizon);
        new Arrival ().schedule (genArr.nextDouble ());
        Sim.start ();
    }

    class Arrival extends Event {
        public void actions () {
            // Next arrival.
            new Arrival ().schedule (genArr.nextDouble ());
            Customer cust = new Customer (); // Cust just arrived.
            cust.arrivTime = Sim.time ();
            cust.servTime = genServ.nextDouble ();
            if (servList.size () > 0) { // Must join the queue.
                waitList.addLast (cust);
                totWait.update (waitList.size ());
            } else { // Starts service.

```

```

        custWaits.add (0.0);
        servList.addLast (cust);
        new Departure ().schedule (cust.servTime);
    }
}

class Departure extends Event {
    public void actions () {
        servList.removeFirst ();
        if (waitList.size () > 0) {
            // Starts service for next one in queue.
            Customer cust = (Customer)waitList.removeFirst ();
            totWait.update (waitList.size ());
            custWaits.add (Sim.time () - cust.arrivTime);
            servList.addLast (cust);
            new Departure ().schedule (cust.servTime);
        }
    }
}

class EndOfSimulation extends Event {
    public void actions () {
        Sim.stop ();
    }
}

public static void main (String [] args) {
    QueueEv queue = new QueueEv (1.0/10.0, 1.0/9.0);
    queue.simulateOneRun (1000.0);
    System.out.println (queue.custWaits.report ());
    System.out.println (queue.totWait.report ());
}
}

```

### 3.2.1 Simulation sans liste d'événements.

Il n'est pas toujours indispensable de recourir à l'approche par événements. A titre d'exemple, considérons la récurrence de Lindley :

$$W_1 = 0, \quad W_{i+1} = \max(0, W_i + S_i - A_{i+1}).$$

Nous pouvons ainsi facilement simuler pour un nombre fixe de clients (au lieu d'un horizon de temps  $T$  fixe).

```

public class QueueLindley {
    RandomStream streamArr = new RandMrg ();

```

```

RandomStream streamServ = new RandMrg();

public double simulateOneRun (int numCust, double lambda, double mu) {
    double Wi = 0.0;
    double sumWi = 0.0;
    for (int i = 2; i <= numCust; i++) {
        Wi += ExponentialDist.inverseF (mu, streamServ.nextDouble()) -
            ExponentialDist.inverseF (lambda, streamArr.nextDouble());
        if (Wi < 0.0) Wi = 0.0;
        sumWi += Wi;
    }
    return sumWi / numCust;
}

public static void main (String[] args) {
    System.out.println ("Average waiting time: " +
        (new QueueLindley()).simulateOneRun (100, 1.0/10.0, 1.0/9.0));
}
}

```

Le formulation du programme basée sur la récurrence de Lindsley revient à traiter un problème d'intégration sur  $[0, 1)^t$ . Supposons que nous souhaitions estimer  $E[\overline{W}_{100}]$  par  $\overline{W}_{100}$ . Il nous faut par conséquent pouvoir calculer  $W_1, \dots, W_{100}$ . Autrement dit, il nous faut  $A_1, S_1, A_2, S_2, \dots, A_{99}, S_{99}$ . Si on pose  $A_i = F^{-1}(U_{2i-1})$  et  $S_i = G^{-1}(U_{2i})$ , où  $U_j, j = 1, \dots, 99$ , sont des variables aléatoires uniformes sur  $(0, 1)$ , alors  $\overline{W}_{100} = f(U_1, \dots, U_{198})$  ( $f$  étant de forme inconnue dans le cas présent). Nous avons donc  $t = 198$ , i.e.,

$$E[\overline{W}_{100}] = \int_{[0,1]^{198}} f(\mathbf{u}) d\mathbf{u}.$$

Par contre, si nous voulons simuler  $\overline{W}_{N_c(T)}$ , le nombre de clients  $N_c(T)$  est aléatoire et non borné. Nous devrions donc choisir  $t = \infty$  comme dimension.

### 3.3 Approche par processus

Un processus est une séquence temporellement ordonnée d'événements inter-reliés, séparés par certains intervalles de temps, qui décrit l'expérience entière d'une "entité" comme celle-ci évolue à travers un "système". Un système ou un modèle de simulation peut avoir différents types de processus. Une routine sera associée à chaque processus du modèle, décrivant son histoire entière à travers le système. Au contraire de l'approche par événements, une routine de processus

contient explicitement le passage du temps simulé et a généralement de multiples points d'entrée.

Une simulation utilisant l'approche par processus évolue aussi au cours du temps en exécutant les événements dans l'ordre de leur occurrence. En interne, les approches par événement et par processus sont donc très similaires, bien que pour l'utilisateur, le raisonnement soit différent, et puisse parfois (mais pas toujours) être plus naturel pour l'approche par processus. En outre, l'approche par processus est souvent plus lente en terme de temps d'exécution, puisque le simulateur doit au final traiter des événements, et gérer des processus pouvant être concurrents.

**Exemple 10** (Simulation par processus d'une file M/M/1). *Dans une simulation par événements, chaque client est passif, et n'effectue rien par lui-même. Il ne sert qu'à stocker deux nombres : le temps d'attente et le temps de service. Dans une approche de programmation par processus, chaque client est vu comme un processus. Le serveur devient une ressource utilisée par le client, comme illustré dans la Figure 3.3.*

*En SSJ, nous pourrions écrire le programme de simulation comme suit :*

```
public class QueueProc {

    Resource server = new Resource (1, "Server");
    RandomVariateGen genArr;
    RandomVariateGen genServ;

    public QueueProc (double lambda, double mu) {
        genArr = new RandomVariateGen (new RandMrg(),
            new ExponentialDist (lambda));
        genServ = new RandomVariateGen (new RandMrg(),
            new ExponentialDist (mu));
    }

    public void simulateOneRun (double timeHorizon) {
        SimProcess.init();
        server.setStatCollecting (true);
        new EndOfSimulation().schedule (timeHorizon);
        new Customer().schedule (genArr.nextDouble());
        Sim.start();
    }

    class Customer extends SimProcess {
        public void actions() {
            new Customer().schedule (genArr.nextDouble());
            server.request (1);
            delay (genServ.nextDouble());
            server.release (1);
        }
    }
}
```

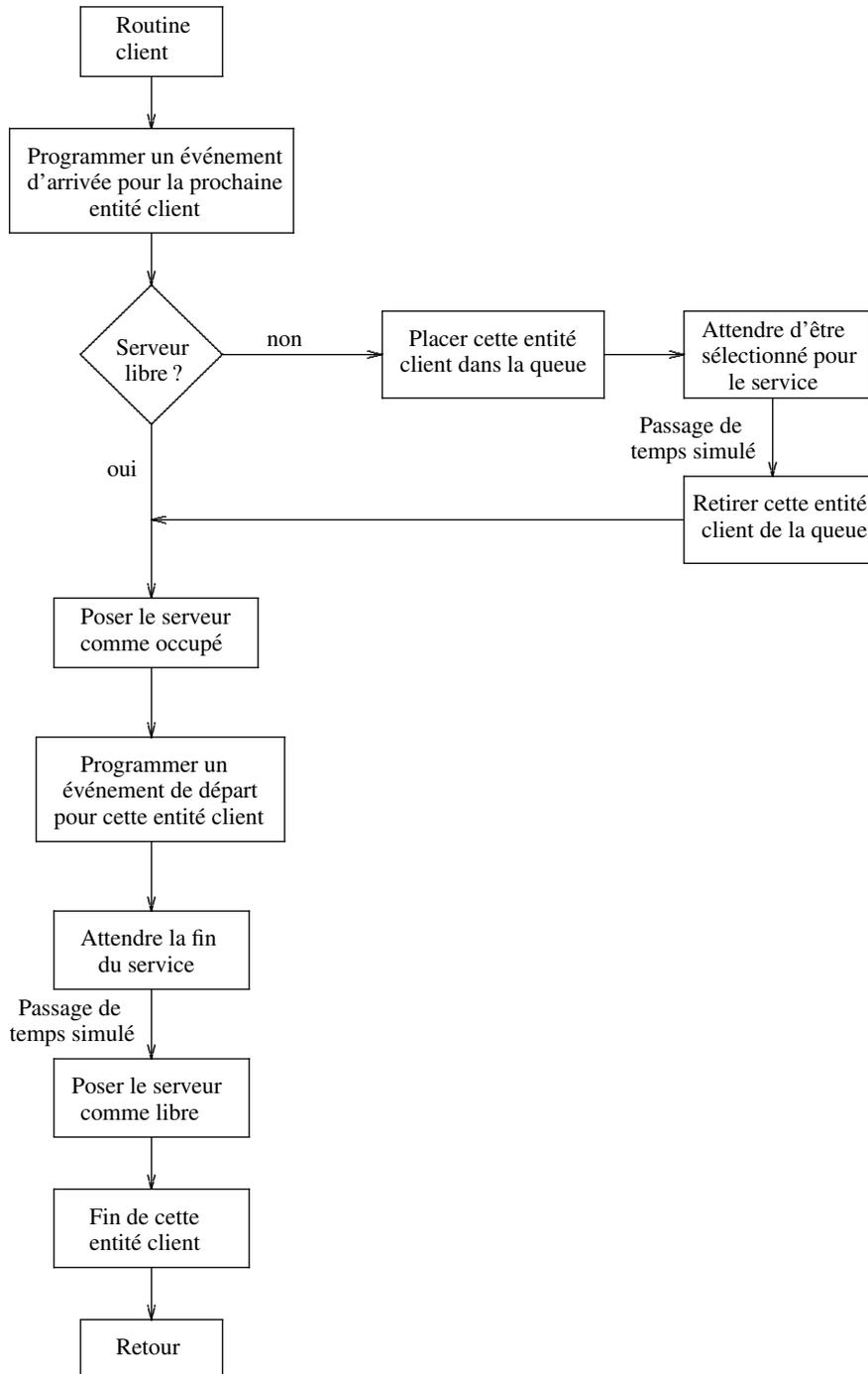


FIGURE 3.3 – Approche par processus

```

    }
}

class EndOfSimulation extends Event {
    public void actions() {
        Sim.stop();
    }
}

public static void main (String[] args) {
    QueueProc queue = new QueueProc (1.0/10.0, 1.0/9.0);
    queue.simulateOneRun (1000.0);
    System.out.println (queue.server.report());
}
}

```

Résultats du programme *QueueProc* :

```

REPORT ON RESOURCE : Server
  From time : 0.0   to time : 1000.0

```

	min	max	average	std. dev.	num. obs.
Capacity	1	1	1		
Utilisation	0	1	0.999		
Queue Size	0	12	4.850		
Wait	0	113.721	49.554	22.336	97
Service	0.065	41.021	10.378	10.377	96
Sojourn	12.828	124.884	60.251	21.352	96

$N_c$  désigne ici le nombre de clients ayant été servis ( $N_c(1000) = 96$ ) entre le début et l'horizon de la simulation. Attention : les 97 observations  $W_i$  ici ne sont pas indépendantes ! Si on veut calculer un intervalle de confiance pour  $w_T = E[\bar{W}_{N_c(T)}]$ , il faut faire plusieurs répétitions indépendantes de cette simulation. Nous pourrions par exemple exécuter huit fois la simulation dans *QueueProc*, avec  $T = 1000$  :

```

public static void main (String[] args) {
    QueueProc q = new QueueProc ();
    for (int i = 0; i < 8; i++)
        System.out.println ("Average waiting time: " +
            q.simulateOneRun (100, 1.0/10.0, 1.0/9.0));
}

```

Résultats avec  $T = 1000$  :

<i>Essai</i>	$N_c$	$\bar{W}_{N_c}$	$\bar{Q}_T$
1	97	49.55	4.85
2	95	27.72	2.64
3	92	44.02	4.93
4	101	27.91	2.82
5	88	27.43	2.41
6	95	60.07	8.40
7	102	20.92	2.49
8	84	24.91	2.09

Puis avec  $T = 10000$  :

<i>Essai</i>	$N_c$	$\bar{W}_{N_c}$	$\bar{Q}_T$
1	1033	124.52	12.87
2	997	82.93	8.27
3	998	72.40	7.31
4	1003	45.79	4.59
5	1016	82.67	8.40
6	1003	65.30	6.56
7	974	91.16	9.36
8	976	62.76	6.14

Lorsque  $T$  augmente, on note empiriquement que la variance semble diminuer et que les moyennes semblent plus élevées.

Valeurs théoriques sur horizon infini : pour un modèle M/M/1, on connaît les moyennes pour  $T \rightarrow \infty$ . Supposons un taux d'arrivée  $\lambda = 1/10$  et un taux de service  $\mu = 1/9$ . En vertu de (C.1), le facteur d'utilisation est donné par  $\rho = \lambda/\mu = 0.9$ . Sur base de la formule de Pollaczek-Khintchine (2.1), le temps d'attente moyen dans la file vaut

$$w \stackrel{\text{def}}{=} \lim_{T \rightarrow \infty} E[\bar{W}_{N_c(T)}] = q/\lambda = \rho^2/((1-\rho)\lambda) = 81.$$

Le nombre moyen de clients dans la file par conséquent

$$q \stackrel{\text{def}}{=} \lim_{T \rightarrow \infty} E[\bar{Q}_T] = \lambda w = \rho^2/(1-\rho) = 8.1.$$

Si nous comparons avec les résultats de la simulation, nous observons un biais. Pourquoi ? Le système est initialement vide et  $W_1 = 0$ . Par contre, les estimateurs sont consistants lorsque  $T \rightarrow \infty$ .

*Echauffement de la simulation : nous pouvons diminuer le biais en recueillant les statistiques seulement à partir d'un certain temps  $T_0$ . Les estimateurs deviennent :*

$$\frac{1}{N_c(T) - N_c(T_0)} \sum_{i=N_c(T_0)+1}^{N_c(T)} W_i$$

et

$$\frac{1}{T - T_0} \int_{T_0}^T Q(t) dt.$$

*Comment choisir  $T_0$  ?*

*Question très difficile en général.*

*Compromis entre le biais et la variance.*

*Problème 1 : variance élevée.*

*Remèdes possibles :*

- Plus grand nombre de répétitions ; prendre la moyenne. (Bien sûr, il faut aussi calculer un intervalle de confiance.)
- Utiliser un horizon plus long.
- Utiliser des techniques de réduction de la variance.

*Problème 2 : Biais dû à l'état initial.*

*Danger : attention aux intervalles de confiance !*

*Améliorations possibles :*

- Utiliser un horizon plus long.
- Echauffement.

## 3.4 Exercices

1. Prouvez la validité de la récurrence de Lindley.



## Chapitre 4

# Générateurs uniformes $[0, 1]$

### 4.1 Principes de base

Le principe de base d'un générateur de nombres aléatoires uniformes est de définir une fonction de transition  $f : \mathcal{S} \rightarrow \mathcal{S}$ , où  $\mathcal{S}$  est l'espace d'état, de cardinalité finie. L'état initial est dénoté par  $s_0$ , et nous écrirons

$$s_n = f(s_{n-1}).$$

Supposons de plus que  $f$  est périodique pour tout  $n$  plus grand ou égal à un certain  $\tau$  connu (souvent égal à 0), de période  $\rho$ , inférieure ou égale à la cardinalité de  $\mathcal{S}$ . En d'autres termes, nous avons  $s_{n+\rho} = s_n$  pour tout  $n \geq \tau$ . Nous dénoterons l'espace de sortie par  $\mathcal{U}$ , et nous supposerons dans le présent document que  $\mathcal{U} = (0, 1)$ . Le fonction de sortie  $g : \mathcal{S} \rightarrow \mathcal{U}$  transforme l'état  $s_n$  dans la valeur de sortie  $u_n$ . Ce comportement est illustré graphiquement dans la Figure 4.1, avec  $\tau = 0$ .

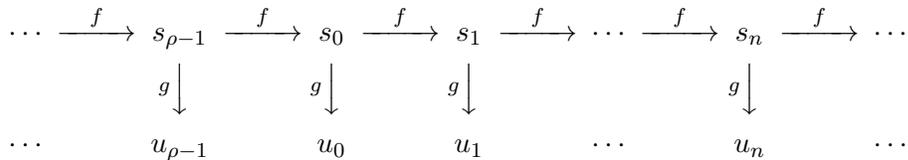


FIGURE 4.1 – Comportement d'un générateur de nombres aléatoires

L'objectif est de rendre difficile, en observant seulement  $(u_0, u_1, \dots)$ , de distinguer cette suite d'une suite de variables aléatoires i.i.d. uniformes sur  $\mathcal{U}$ , bien

que la suite n'ait rien d'aléatoire : une fois l'état initial fixé, l'entièreté de la suite est fixée de manière déterministe, comme souligné par exemple par John von Neumann [18] :

”Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin. For, as has been pointed out several times, there is no such things as a random number—there are only methods to produce random numbers, and a strict arithmetic procedure of course is not such a method. [...] We are here dealing with mere “cooking recipes” for making digits[.]”

Un générateur de nombres aléatoires peut être résumé en citant Lehmer [10] :

”A random sequence is a vague notion embodying the idea of a sequence in which each term is unpredictable to the uninitiated and whose digits pass a certain number of tests traditional with statisticians and depending somewhat on the use to which the sequence is to be put.”

L'histoire des générateurs de nombres (pseudo)-aléatoires débuta véritablement en 1946, avec l'introduction par von Neumann de la méthode du carré médian, malheureusement peu performante. Très simple, elle consiste à prendre un nombre, à l'élever au carré et à prendre les chiffres au milieu comme sortie.

## 4.2 Générateurs récursifs

### 4.2.1 Générateur congruentiel linéaire (GCL)

Dès 1948 (Lehmer [10]) furent introduits des générateurs de la forme

$$ax + c \pmod{m}. \quad (4.1)$$

En supposant tout d'abord que  $c$  vaut 0 (comme dans l'approche proposée par Lehmer), la période maximale est  $m - 1$  et est atteinte si et seulement si  $m$  est premier et  $a$  est une racine primitive de  $m$ .  $r$  est une racine primitive de  $m$  si les puissances de  $r$  ( $1, r, r^2, r^3, \dots$ ) génèrent tous les entiers non-nuls modulo  $m$ . Puisqu'il y a  $m - 1$  entiers non nuls, ceci signifie que les premières  $m - 1$  puissances de  $r$  doivent être différentes, modulo  $m$ . De manière équivalente, nous pouvons parler de l'ordre de  $r$ . L'ordre d'une racine  $r$  de  $m$  est le plus petit entier (strictement) positif  $x$  tel que  $r^x = 1 \pmod{m}$ .  $r$  est une racine primitive si son ordre est  $m - 1$ . Il est possible de montrer que ceci équivaut à exiger que  $a^{(m-1)/p} - 1$  est un multiple de  $m$  pour chaque facteur premier  $p$  de  $m - 1$ , ou encore le plus petit entier  $l$

pour lequel  $r^l - 1$  est divisible par  $m$  est  $l = m - 1$ . Les générateurs congruentiels linéaires qui remplissent ces conditions sont appelés GCL's multiplicatifs à modulus premier. Notons que la condition  $m$  premier suffit pour garantir l'existence d'un générateur de période maximale, en vertu du théorème ci-dessous.

**Théorème 1.** *Si  $m$  est premier, il existe une racine primitive pour  $m$ .*

Il n'existe malheureusement pas de méthode simple pour calculer ces racines.

**Exemple 11.** *Si  $m = 7$ , alors 3 est une racine primitive de  $m$  car les puissances de 3 modulo 7 sont 1, 3, 2, 6, 4, 5, c'est-à-dire chaque entier strictement compris entre 0 et 7. Mais 2 n'est pas une racine primitive de  $m$  car les puissances de 2 modulo 7 sont 1, 2, 4, 1, 2, 4, 1, 2, 4, ...*

Si  $c \neq 0$ , il est possible d'obtenir une période égale à  $m$ , sous les conditions exposées dans le théorème ci-dessous.

**Théorème 2.** *Le GCL (4.1) a une période pleine si et seulement si les trois conditions suivantes tiennent :*

1. *le seul entier positif qui divise de manière exacte à la fois  $m$  et  $c$  est 1 ;*
2. *si  $q$  est un nombre premier qui divise  $m$ , alors  $q$  divise  $a - 1$  ;*
3. *si 4 divise  $m$ , alors 4 divise  $a - 1$ .*

**Exemple 12.** *Park et Miller [13] ont proposé un générateur standard qu'ils ont appelé le Standard Minimal, après avoir testé divers générateurs connus au moment de leur étude. Bien qu'il suffise pour les applications simples, les générateurs présentés dans les sections suivantes le surpassent largement, et par conséquent, il est déconseillé de l'utiliser pour des simulations complexes. Le Standard Minimal est un générateur congruentiel linéaire défini par la récurrence*

$$x_{n+1} = 16807x_n \pmod{(2^{31} - 1)}.$$

#### 4.2.2 Implantation de générateurs congruentiels linéaires

Une difficulté principale est de calculer  $ax \pmod{m}$  pour de grands  $m$ , ce qui entraîne des risques de débordement de registres.

**Factorisation approximative.**

Cette méthode est valide si  $a^2 < m$  ou  $a = \lfloor m/i \rfloor$ , avec  $i^2 < m$ , et procède par des calculs sur des entiers. Précalculons  $q = \lfloor m/a \rfloor$  et  $r = m \bmod a$ , puis

$$\begin{aligned} y &= \lfloor x/q \rfloor; \\ x &= a(x - yq) - yr. \end{aligned}$$

Si  $x < 0$ , nous posons  $x := x + m$ . Justification :

$$\begin{aligned} ax \bmod m &= (ax - \lfloor x/q \rfloor m) \bmod m \\ &= (ax - \lfloor x/q \rfloor (aq + r)) \bmod m \\ &= (a(x - \lfloor x/q \rfloor q) - \lfloor x/q \rfloor r) \bmod m \\ &= (a(x \bmod q) - \lfloor x/q \rfloor r) \bmod m. \end{aligned}$$

Sous les conditions posées, il est immédiat de noter que toutes les quantités intermédiaires demeurent entre  $-m$  et  $m$ . En C, la procédure peut s'exprimer comme suit :

```
long q, r, y;

q = m/a;
r = m%a;

y = x/q;
x = a*(x-y*q)-y*r;

if (x < 0) x += m;
```

**Calculs en point flottant, double précision.**

La procédure est valide si tous les entiers à considérer peuvent être représentés de manière exacte en passant en calcul flottant. En particulier, si la double précision fait appel à 64 bits, et suit la norme IEEE, la procédure suivante est correcte si  $am < 2^{53}$  :

```
double m, a, x, y; int k;
y = a * x; k = [y/m]; x = y - k * m;
```

**Décomposition en puissances de 2.**

Supposons que  $a = \pm 2^q \pm 2^r$  et  $m = 2^e - h$  pour  $h$  petit. Dans ce cas,

$$ax \pmod m = \pm 2^q x \pmod m + \pm 2^r x \pmod m.$$

Pour calculer  $y = 2^q x \pmod m$  (le calcul de  $2^r x$  est similaire), nous décomposons  $x$  en  $x_0 + 2^{e-q}x_1$ , comme illustré sur la Figure 4.2.

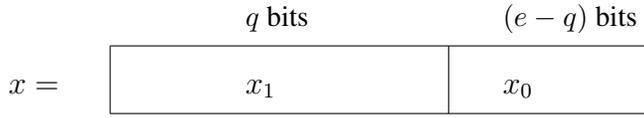


FIGURE 4.2 – Décomposition de l'entier  $x$

Pour  $h = 1$  (Wu, 1997), on obtient  $y$  en permutant  $x_0$  et  $x_1$ . En effet,

$$\begin{aligned} 2^q x \pmod m &= 2^q(x_0 + 2^{e-q}x_1) \pmod{(2^e - 1)} \\ &= 2^q x_0 + [2^e x_1 \pmod{(2^e - 1)}] \\ &= 2^q x_0 + x_1. \end{aligned}$$

Pour  $h > 1$  (L'Ecuyer et Simard 1999), nous avons de la même manière

$$y = 2^q(x_0 + 2^{e-q}x_1) \pmod{(2^e - h)} = (2^q x_0 + h x_1) \pmod{(2^e - h)}.$$

Si  $h < 2^q$  et  $h(2^q - (h + 1)2^{-e+q}) < m$ , comme  $x_0 \leq 2^{e-q}$  et  $x_1 \leq 2^q$ , nous avons

$$2^q x_0 \leq 2^e - 2^q < m.$$

De plus, étant donné que  $2^{e-q}x_1 \leq m - 1$ , nous avons

$$h x_1 \leq h(m - 1)/2^{e-q} = h(2^e - h - 1)/2^{e-q} = h(2^q - (h + 1)2^{-e+q}) < m,$$

et par conséquent chaque terme est strictement inférieur à  $m$ . L'opération modulo revient dès lors à soustraire  $m$  si la somme est  $\geq m$ .

**Exemple 13.** *Le listing suivant illustre comment implanter cette technique en C pour  $m = 2^{30} - 35$  et  $a = 2^{15} + 2^{13}$ .*

```

#define m      1073741789    /* 2^30 - 35 */
#define h      35
#define q      15
#define emq    15           /* e - q */
#define mask1  32767        /* 2^(e-q) - 1 */
#define r      13
#define emr    17           /* e - r */
#define mask2  131071      /* 2^(e-r) - 1 */
#define norm   1.0/m

long x;

double axmodm () {
    unsigned long k, x0, x1;

    x0 = x & mask1;
    x1 = x >> emq;
    k = (x0 << q) + h*x1;

    x0 = x & mask2;
    x1 = x >> emr;
    k += (x0 << r) + h*x1;

    if (k < m) x = k;
    else if (k < 2*m) x = k-m;
    else x = k - 2*m;

    return x*norm;
}

```

L'Ecuyer et Simard ont toutefois démontré que ces générateurs présentent des faiblesses statistiques s'ils sont utilisés de manière directe.

### 4.2.3 Générateur récursif multiple (MRG)

Nous pouvons généraliser la récurrence 4.1 en considérant la récurrence de base :

$$x_n = (a_1x_{n-1} + \dots + a_kx_{n-k}) \bmod m, \quad u_n = x_n/m. \quad (4.2)$$

En pratique, on prendra plutôt  $u_n = (x_n + 1)/(m + 1)$ , ou encore  $u_n = x_n/(m + 1)$  si  $x_n > 0$  et  $u_n = m/(m + 1)$  sinon, mais la structure demeure essentiellement la même. Si  $k = 1$ , nous retrouvons le générateur à congruence linéaire classique, avec  $c = 0$ .

L'état à l'étape  $n$  est  $s_n = \mathbf{x}_n = (x_{n-k+1}, \dots, x_n)^T$ . Espace d'états :  $\mathcal{Z}_m^k$ , de cardinalité  $m^k$ . La période maximale est  $\rho = m^k - 1$ , pour  $m$  premier. On associe au MRG le polynôme caractéristique :

$$P(z) = z^k - a_1 z^{k-1} - \dots - a_k = - \sum_{j=0}^k a_j z^{k-j},$$

où  $a_0 = -1$ .

Pour  $k > 1$ , pour avoir une période maximale, il est possible de montrer qu'il suffit d'avoir au moins deux coefficients non nuls, dont  $a_k$ . Ainsi, la récurrence la plus économique a la forme :

$$x_n = (a_r x_{n-r} + a_k x_{n-k}) \pmod{m},$$

avec  $0 < r < k$ .

Une erreur fréquente, commise en particulier par les informaticiens peu au fait des statistiques, est de considérer  $m = 2^e$ . Utiliser une puissance de 2 pour  $m$  permet en effet de facilement calculer le produit  $ax \pmod{m}$ , et est parfois décrit comme efficace, ce qui est vrai du point de la rapidité d'exécution. Les effets sur la période sont pourtant dommageables, vu que

- pour  $k = 1$  et  $e \geq 4$ , on a  $\rho \leq 2^{e-2}$  ;
- pour  $k > 1$ , on a  $\rho \leq (2^k - 1)2^{e-1}$ .

**Exemple 14.** Si  $k = 7$  et  $m = 2^{31} - 1$ , la période maximale est  $(2^{31} - 1)^7 - 1 \approx 2^{217}$ . Mais pour  $m = 2^{31}$  on a  $\rho \leq (2^7 - 1)2^{31-1} < 2^{37}$ , i.e.  $2^{180}$  fois plus petit !

Pire, si nous nous intéressons au  $i^{\text{th}}$  bit le moins significatif, pour  $k = 1$ , la période de  $x_n \pmod{2^i}$  ne peut pas dépasser  $\max(1, 2^{i-2})$ . Pour  $k > 1$ , la période de  $x_n \pmod{2^i}$  ne peut pas dépasser  $(2^k - 1)2^{i-1}$ .

**Exemple 15.** Récurrence  $x_n = 10205x_{n-1} \pmod{2^{15}}$  :

$$\begin{aligned} x_0 &= 12345 &= 011000000111001_2 \\ x_1 &= 20533 &= 101000000110101_2 \\ x_2 &= 20673 &= 101000011000001_2 \\ x_3 &= 7581 &= 001110110011101_2 \\ x_4 &= 31625 &= 111101110001001_2 \\ x_5 &= 1093 &= 000010001000101_2 \\ x_6 &= 12945 &= 011001010010001_2 \\ x_7 &= 15917 &= 011111000101101_2. \end{aligned}$$

$m$	$a$	$c$	Source
$2^{24}$	1140671485	12820163	early MS VisualBasic
$2^{31}$	65539	0	RANDU (IBM)
$2^{31}$	134775813	1	early Turbo Pascal
$2^{31}$	1103515245	12345	rand() in BSD ANSI C
$2^{32}$	69069	1	VAX/VMS systems
$2^{32}$	2147001325	715136305	BCLP language
$2^{35}$	$5^{15}$	7261067085	Knuth (1998)
$2^{48}$	68909602460261	0	Fishman (1990)
$2^{48}$	25214903917	11	Unix's rand48()
$2^{48}$	44485709377909	0	CRAY system
$2^{59}$	$13^{13}$	0	NAG Fortran/C library

$$x_n = (ax_{n-1} + c) \bmod m; \quad u_n = x_n/m.$$

TABLE 4.1 – générateurs avec  $m = 2^e$ 

De tels générateurs restent malheureusement populaires, comme illustré dans la Table 4.2.3, mais sont à proscrire dans des simulations dignes de ce nom. Ainsi, la fonction `rand48` reste présente dans les bibliothèques C standards BSD.

### 4.3 Générateurs à sous-suites multiples

Afin de pouvoir adéquatement représenter les différentes variables aléatoires, il peut être intéressant de pouvoir instancier des générateurs de variables aléatoires à volonté, et faire évoluer ceux-ci en parallèle, plutôt que d'utiliser un seul générateur et transformer les tirs dans les distributions voulues à la volée. Un tel point est en particulier important pour gérer adéquatement les questions de synchronisation lors de l'utilisation de variables aléatoires communes (voir Section 7.3). Par conséquent, nous voudrions pouvoir utiliser plusieurs fois un même générateur au sein d'un programme, mais en débutant avec des semences différentes afin de produire des suites aléatoires différentes. Une première approche consiste à créer plusieurs générateurs, en spécifiant manuellement ces semences. Le danger majeur de cette approche est qu'il est difficile de prévoir la position de ces semences dans la séquence aléatoire, ce qui peut conduire à produire des séquences fortement corrélées. Le risque est d'autant plus élevé que la période du générateur est faible.

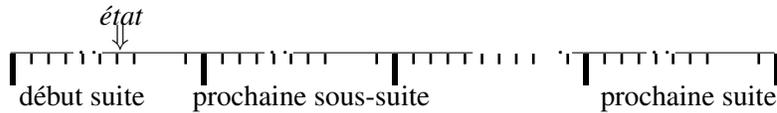


FIGURE 4.3 – Sous-suites d’un générateur de nombres aléatoires.

Dans ce cas, il est ainsi utile de pouvoir partitionner ces suites (ou “streams”) en sous-suites, comme illustré sur la Figure 4.3.

**Exemple 16.** Soit  $X, Y$ , deux variables aléatoires normales  $N(0, 1)$  indépendantes. Il est possible de montrer que le rapport  $X/Y$  suit une distribution de Cauchy (voir Section A.4.7). Si nous générons ce rapport à l’aide du générateur congruentiel linéaire Standard Minimal, en prenant comme semence 1 pour le numérateur, et 2 pour le dénominateur, nous obtenons la distribution empirique représentée sur la Figure 4.4, qui est clairement très éloignée de la distribution recherchée.

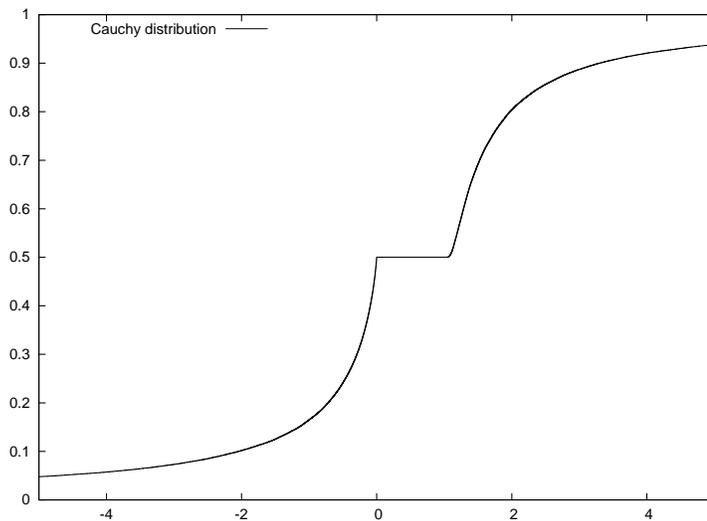


FIGURE 4.4 – distribution de Cauchy erronément générée

**Exemple 17.** Tout générateur présent dans la librairie SSJ implémentera l’interface générale ci-dessous.

```

public interface RandomStream {
    // Réinitialise la suite à son état initial.
    public void resetStartStream ();
    // Réinitialise la suite au début de sa sous-suite courante.
    public void resetStartSubstream ();
    // Réinitialise la suite au début de sa prochaine sous-suite.
    public void resetNextSubstream ();
    // Retourne une v.a. U(0,1) de cette suite et avance d'un pas.
    public double nextDouble ();
    // Retourne une v.a. uniforme sur {i,i+1,...,j}.
    public int nextInt (int i, int j);
}

```

*Nous aurons ainsi une déclaration du type*

```
public class RandMrg implements RandomStream
```

*et une instance particulière peut être créée à l'aide du constructeur RandMrg ()*

*La méthode*

```
public static void setPackageSeed (long seed[]);
```

*fixe quant à elle l'état initial de la première suite. Les autres états initiaux seront  
sont calculés selon un espacement prédéterminé.*

Pour passer d'une suite à une autre, il est nécessaire de pouvoir calculer un point de la récurrence sans devoir générer tous les points intermédiaires. Or, nous pouvons écrire

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} \pmod m = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ a_k & a_{k-1} & \cdots & a_1 \end{pmatrix} \mathbf{x}_{n-1} \pmod m.$$

Ainsi

$$\mathbf{x}_{n+\nu} = \mathbf{A}^\nu \mathbf{x}_n \pmod m = (\mathbf{A}^\nu \pmod m) \mathbf{x}_n \pmod m.$$

Nous pouvons précalculer  $\mathbf{A}^\nu \pmod m$  au moyen de la procédure suivante :

$$\mathbf{A}^\nu \pmod m = \begin{cases} (\mathbf{A}^{\nu/2} \pmod m)(\mathbf{A}^{\nu/2} \pmod m) \pmod m & \text{si } \nu \text{ est pair;} \\ \mathbf{A}(\mathbf{A}^{\nu-1} \pmod m) \pmod m & \text{si } \nu \text{ est impair.} \end{cases}$$

## 4.4 Structure de réseau

Considérons le sous-ensemble de  $[0, 1]^t$  construit à partir des différents états initiaux possibles du générateur.

$$\Psi_t = \{\mathbf{u} = (u_0, \dots, u_{t-1}) = (g(s_0), \dots, g(s_{t-1})), s_0 \in \mathcal{S}\}.$$

Un critère majeur est que  $\Psi_t$  doit recouvrir  $[0, 1]^t$  *très uniformément*, et ce pour “tout”  $t$ . Par généralisation, nous chercherons également à mesurer l’uniformité de  $\Psi_I = \{(u_{i_1}, \dots, u_{i_t}) \mid s_0 \in \mathcal{S}\}$  pour une classe choisie d’ensembles d’indices de forme  $I = \{i_1, i_2, \dots, i_t\}$ . La récurrence linéaire à la base d’un MRG a comme conséquence majeure de produire une structure pour l’ensemble  $\Psi_t$ , comme illustré sur les Figures 4.5 à 4.7.

Pour ce faire, considérons un vecteur quelconque  $(x_0, \dots, x_{k-1})$  dans  $\{0, 1, \dots, m-1\}^k$ , et la base canonique de  $\mathbb{R}^k$  :

$$\{\mathbf{e}_i, i = 1, \dots, k\},$$

où  $\mathbf{e}_i$  est un vecteur dont les composantes sont toutes nulles, sauf pour la  $i^{\text{e}}$ , laquelle vaut 1. Si  $(x_0, \dots, x_{k-1}) = \mathbf{e}_1 = (1, 0, \dots, 0)$ , la récurrence du MRG donne

$$\begin{aligned} (x_1, \dots, x_k) &= (0, \dots, a_k), \\ (x_2, \dots, x_k, x_{k+1}) &= (0, \dots, a_k, a_1 a_k \pmod{m}), \\ (x_3, \dots, x_{k+2}) &= (0, \dots, (a_1^2 + a_2) a_k \pmod{m}), \dots \end{aligned}$$

Si  $(x_0, \dots, x_{k-1}) = \mathbf{e}_2 = (0, 1, \dots, 0)$ , alors

$$\begin{aligned} (x_1, \dots, x_k) &= (1, 0, \dots, a_{k-1}), \\ (x_2, \dots, x_k, x_{k+1}) &= (0, \dots, a_{k-1}, (a_1 a_{k-1} + a_k) \pmod{m}), \\ (x_3, \dots, x_{k+2}) &= (0, \dots, (a_1^2 a_{k-1} + a_1 a_k + a_2 a_{k-1}) \pmod{m}), \dots \end{aligned}$$

Nous pouvons continuer de la sorte jusqu’à considérer  $(x_0, \dots, x_{k-1}) = \mathbf{e}_k = (0, \dots, 0, 1)$ , ce qui produit

$$\begin{aligned} (x_1, \dots, x_k) &= (0, \dots, 1, a_1), \\ (x_2, \dots, x_k, x_{k+1}) &= (0, \dots, 1, a_1, (a_1^2 + a_2) \pmod{m}), \dots \end{aligned}$$

Or tout vecteur  $(x_n, \dots, x_{n+t-1})$  qui obéit à la récurrence, pour  $t \geq k$ , est une combinaison linéaire à coefficients entiers de ces  $k$  vecteurs de base. Pour le voir, notons  $x_{i,0}, x_{i,1}, x_{i,2}, \dots$  la suite obtenue à partir du vecteur de base  $\mathbf{e}_i$ . Un état initial  $(x_0, \dots, x_{k-1}) = (z_1, \dots, z_k)$  peut s’écrire comme  $z_1 \mathbf{e}_1 + \dots + z_k \mathbf{e}_k$  et produit la suite  $z_1(x_{1,0}, x_{1,1}, \dots) + \dots + z_k(x_{k,0}, x_{k,1}, \dots) \pmod{m}$ , et réciproquement. La réduction

modulo  $m$  se fait en soustrayant des vecteurs  $m\mathbf{e}_i$ . Ainsi, pour  $t \geq k$ ,  $(x_0, x_1, \dots, x_{t-1})$  suit la récurrence si et seulement si il s'agit d'une combinaison linéaire à coefficients entiers de

$$\begin{aligned} & (1, 0, \dots, 0, x_{1,k}, \dots, x_{1,t-1}) \\ & \quad \vdots \\ & (0, 0, \dots, 1, x_{k,k}, \dots, x_{k,t-1}) \\ & \quad (0, 0, \dots, 0, m, \dots, 0) \\ & \quad \vdots \\ & (0, 0, \dots, 0, 0, \dots, m). \end{aligned}$$

En divisant par  $m$ , on obtient que  $(u_0, \dots, u_{t-1}) \in [0, 1]^t$  est dans  $\Psi_t$  si et seulement si c'est une combinaison linéaire (sur les entiers) de

$$\begin{aligned} \mathbf{v}_1 &= (1, 0, \dots, 0, x_{1,k}, \dots, x_{1,t-1})^T / m \\ & \quad \vdots \\ \mathbf{v}_k &= (0, 0, \dots, 1, x_{k,k}, \dots, x_{k,t-1})^T / m \\ \mathbf{v}_{k+1} &= (0, 0, \dots, 0, 1, \dots, 0)^T \\ & \quad \vdots \\ \mathbf{v}_t &= (0, 0, \dots, 0, 0, \dots, 1)^T. \end{aligned}$$

Si

$$L_t = \left\{ \mathbf{v} = \sum_{i=1}^t z_i \mathbf{v}_i \mid z_i \in \mathcal{Z} \right\}$$

est le réseau ayant ces vecteurs pour base, alors  $\Psi_t = L_t \cap [0, 1]^t$ . Les Figures 4.5 à 4.7 illustrent les réseaux obtenus à partir de quelques LCGs.

Pour  $t > k$ , il y a  $m^t$  vecteurs dont les coordonnées sont des multiples de  $1/m$ , mais seulement  $m^k$  sont dans  $\Psi_t$  (il n'y a que  $m^k$  états initiaux possible), soit une proportion de  $1/m^{t-k}$ . Cette structure de réseau implique que les points de  $\Psi_t$  sont distribués suivant un schéma très régulier. Par exemple, chaque point de  $L_t$  a un plus proche voisin à la même distance et dans la même direction que n'importe quel autre point, et il y a des familles d'hyperplans parallèles équidistants qui couvrent tous les points.

Au regard de cette structure particulière, des manières naturelles de mesurer l'uniformité de ce genre d'ensemble de points  $\Psi_t$  incluent :

1. la distance d'un point à son plus proche voisin, qui est aussi la distance de l'origine au point le plus proche, ou de manière équivalente le vecteur non nul le plus court dans  $L_t$  ;

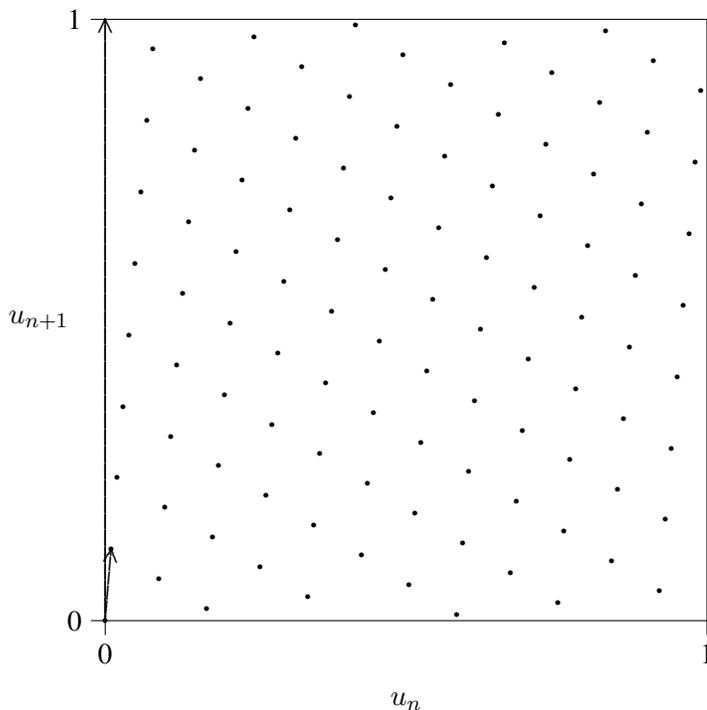


FIGURE 4.5 – LCG avec  $m = 101$  et  $a = 12$ ;  $\mathbf{v}_1 = (1, 12)/101$ ,  $\mathbf{v}_2 = (0, 1)$

2. La distance entre les deux hyperplans les plus éloignés qui couvrent une région ne contenant aucun point de  $L_t$ ;
3. le nombre minimum d'hyperplan équidistants parallèles qui peuvent couvrir tous les points de  $\Psi_t$ .

Pour obtenir la première mesure, nous devons calculer le plus court vecteur non nul dans un réseau de base  $\mathbf{v}_1, \dots, \mathbf{v}_t$  :

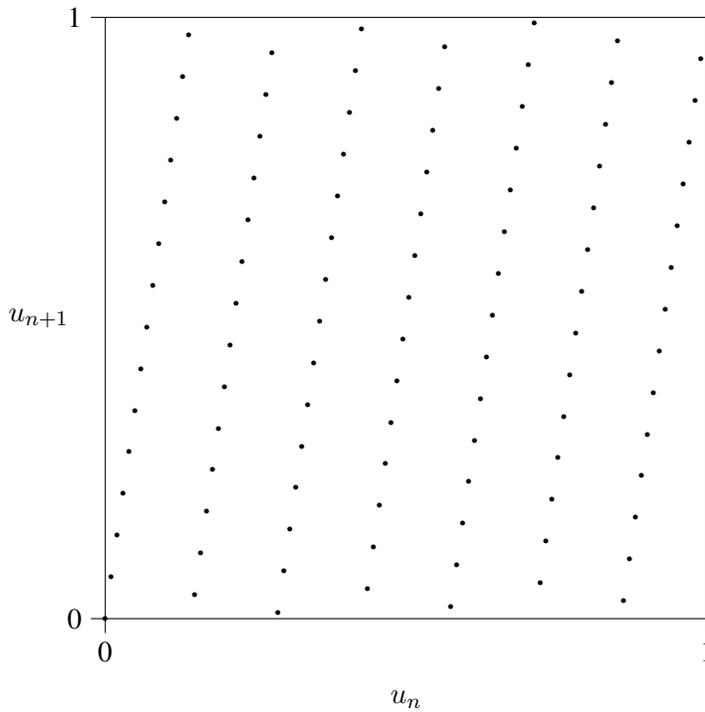
$$\text{Minimiser } \|\mathbf{v}\|_2^2 = \sum_{i=1}^t \sum_{j=1}^t z_i \mathbf{v}_i^T \mathbf{v}_j z_j$$

sous les contraintes que  $z_1, \dots, z_t$  soient entiers et non tous nuls. Il s'agit par conséquent d'un problème d'optimisation quadratique en nombres entiers, lequel est difficile à résoudre.

Pour traiter les deux autres mesures proposées, nous avons tout d'abord besoin de définir le réseau dual, dénoté  $L_t^*$ , comme suit :

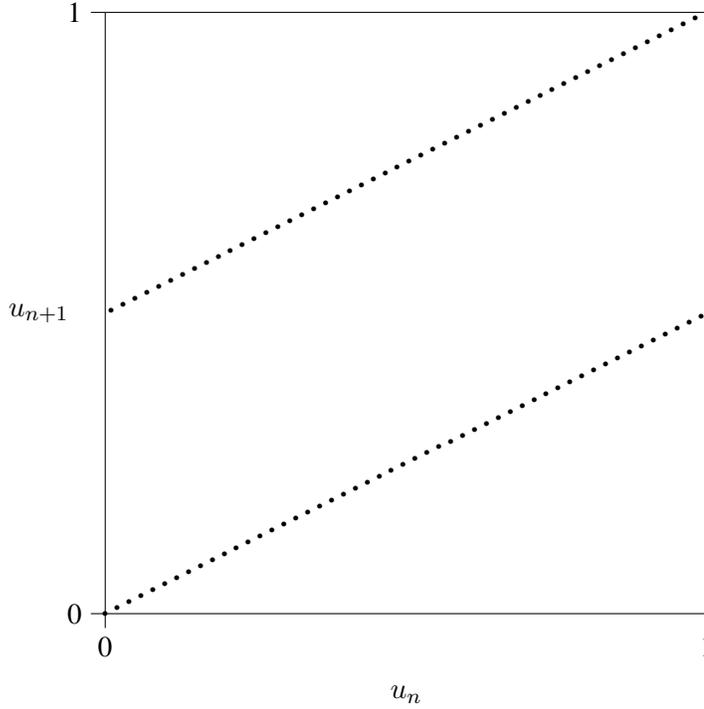
$$L_t^* = \{\mathbf{h} \in \mathcal{R}^t : \mathbf{h} \cdot \mathbf{v} \in \mathcal{Z} \text{ pour tout } \mathbf{v} \in L_t\}.$$



FIGURE 4.6 – LCG with  $m = 101$  and  $a = 7$ 

Pour chaque vecteur  $\mathbf{h} \in L_t^*$  et chaque entier  $z$ , l'ensemble  $\{\mathbf{v} \in \mathcal{R}^t : \mathbf{h}^T \mathbf{v} = z\}$  est un hyperplan orthogonal à  $\mathbf{h}$ . Quand  $z$  parcourt tous les entiers, nous obtenons dès lors une famille d'hyperplans parallèles qui couvrent tous les points  $\mathbf{v}$  de  $L_t$ , car  $\mathbf{h}^T \mathbf{v} \in \mathcal{Z}$  pour  $\mathbf{v} \in L_t$ . La distance entre deux hyperplans est la distance entre l'hyperplan défini avec  $z = 1$  et celui défini avec  $z = 0$ , i.e., à l'origine, puisque ce dernier contient l'origine. Il est possible de montrer que cette distance vaut  $1/\|\mathbf{h}\|_2$ . Si  $\ell_t$  est la longueur du plus court vecteur  $\mathbf{h}$  non nul dans  $L_t^*$ , alors la distance entre les hyperplans pour la famille où ils sont le plus éloignés est  $1/\ell_t$ . Par conséquent, nous cherchons à maximiser  $\ell_t$ .

Un petit nombre d'hyperplans parallèles couvrant tous les points de  $\psi_t$  peut être trouvé en calculant le vecteur non-nul le plus court dans  $L_t^*$  en utilisant la norme  $\mathcal{L}_1$  au lieu de la norme euclidienne. Pour plus de détails, voir Dieter [5].

FIGURE 4.7 – LCG with  $m = 101$  and  $a = 51$ 

#### 4.4.1 Indices lacunaires.

Plutôt que de considérer des indices consécutifs, nous pouvons considérer n'importe quel ensemble de  $t$  indices (distincts)  $I = \{i_1, i_2, \dots, i_t\}$ . Dans ce cas, nous avons

$$\begin{aligned} \Psi_I &= \{(u_{i_1}, \dots, u_{i_t}) \mid s_0 = (x_0, \dots, x_{k-1}) \in \mathcal{Z}_m^k\}, \\ &= L_I \cap [0, 1]^t, \\ 1/\ell_I &= \text{distance entre les hyperplans dans } L_I. \end{aligned}$$

#### 4.4.2 Mesures d'uniformité.

Des bornes supérieures de la forme  $\ell_t \leq \ell_t^*(n)$  pour un réseau général de densité  $n$  dans  $\mathcal{R}^t$  existent. Nous pouvons dès lors standardiser  $\ell_t$  par  $\ell_t/\ell_t^*(m^k)$  pour avoir une mesure dans  $[0, 1]$ , ce qui permet d'obtenir la figure de mérite générale :

$$M_{\mathcal{J}} = \min_{I \in \mathcal{J}} \ell_I / \ell_{|I|}^*(m^k)$$

où  $\mathcal{J}$  est une famille d'ensembles  $I = \{i_1, i_2, \dots, i_t\}$ . La recherche de générateurs potentiellement intéressants passe alors par le calcul numérique des paramètres qui maximisent cette mesure.

## 4.5 Autres bornes.

**Proposition 1.** Si  $i \in I$  lorsque  $a_{k-i} \neq 0$  (avec  $a_0 = -1$ ), alors

$$\ell_I^2 \leq 1 + a_1^2 + \dots + a_k^2.$$

Il faut donc que cette somme soit grande !



*Preuve partielle.* Considérons  $I = \{0, \dots, t-1\}$  et soit  $\mathbf{h} = (-a_k, \dots, -a_1, 1, 0, \dots, 0)^T$ .

Si  $\mathbf{v} = (v_0, v_1, \dots, v_{t-1})^T \in L_t$ , alors  $v_k = (a_1 v_{k-1} + \dots + a_k v_0) \pmod{1}$ .

Ainsi,  $\mathbf{h}^T \mathbf{v}$  doit être un entier, i.e.,  $\mathbf{h} \in L_t^*$ .

Donc  $\ell_t^2 \leq \|\mathbf{h}\|_2^2 = 1 + a_1^2 + \dots + a_k^2$ .

Se généralise au cas de  $\ell_I$ . □

**Exemple 18.** *Lagged-Fibonacci*

$$x_n = (\pm x_{n-r} \pm x_{n-k}) \pmod{m}.$$

Pour  $I = \{0, k-r, k\}$ , on a  $1/\ell_I \geq 1/\sqrt{3} \approx .577$ .

Les vecteurs  $(u_n, u_{n+k-r}, u_{n+k})$  sont tous contenus dans deux plans !

**Exemple 19.** *L'exemple à éviter : la fonction random de la glibc*<sup>1</sup>

Le générateur commence par initialiser un tableau de 34 nombres sur le principe du générateur Standard Minimal :

1.  $x_0 = s$ ;  $x_i = 16807 * x_{i-1} \pmod{2^{31} - 1}$  (pour  $i = 1, \dots, 30$ );

2.  $x_i = x_{i-31}$  (pour  $i = 31, \dots, 33$ ).

Pour  $i \geq 34$ , l'algorithme suit l'algorithme de Lagged-Fibonacci, avec  $m = 2^{31}$  :

$$x_i = (x_{i-3} + x_{i-31}) \pmod{2^{31}}.$$

Enfin, la fonction ignore les 344 premiers nombres et supprime le bit de poids faible :

$$o_i = x_{i+344} \gg 1.$$

---

1. Cette remarque a été écrite sur base de la version 2.9, toujours d'utilisation en 2009, soit 60 ans après l'apparition des générateurs congruentiels linéaires.

En conclusion, on observe que le générateur résume ce qu'il convient de ne pas faire :

1. le générateur Standard Minimal est dépassé ;
2. les vecteurs  $x_i, x_{i+28}, x_{i+31}$  sont contenus dans deux plans ;
3. la linéarité n'est pas complètement supprimée en enlevant le bit de poids faible, vu que pour  $i \geq 34$ ,

$$o_i = o_{i-31} + o_{i-3} \pmod{2^{31}} \text{ ou } o_i = o_{i-31} + o_{i-3} + 1 \pmod{2^{31}},$$

et on conserve la majeure partie des inconvénients liés à  $m = 2^{31}$ .

### 4.5.1 MRGs combinés

Considérons deux [ou plusieurs...] MRGs évoluant en parallèle :

$$\begin{aligned} x_{1,n} &= (a_{1,1}x_{1,n-1} + \cdots + a_{1,k}x_{1,n-k}) \pmod{m_1}, \\ x_{2,n} &= (a_{2,1}x_{2,n-1} + \cdots + a_{2,k}x_{2,n-k}) \pmod{m_2}. \end{aligned}$$

On définit les deux *combinaisons* :

$$\begin{aligned} z_n &:= (x_{1,n} - x_{2,n}) \pmod{m_1}; & u_n &:= z_n/m_1; \\ w_n &:= (x_{1,n}/m_1 - x_{2,n}/m_2) \pmod{1}. \end{aligned}$$

La suite  $\{w_n, n \geq 0\}$  est la sortie d'un autre MRG, de module  $m = m_1m_2$ , et  $\{u_n, n \geq 0\}$  est presque la même suite si  $m_1$  et  $m_2$  sont proches. Peut atteindre la période  $(m_1^k - 1)(m_2^k - 1)/2$ .

Permet d'implanter efficacement un MRG ayant un grand  $m$  et plusieurs grands coefficients non nuls.

Pour accélérer la génération de point, il est possible de prendre tous les  $a_j$  non nuls égaux à  $a$  (Deng et Xu 2002). Alors,  $x_n = a(x_{n-i_1} + \cdots + x_{n-k}) \pmod{m}$ . Une seule multiplication. Les meilleurs générateurs ne jouissent cependant pas de cette propriété.

Tableaux de paramètres : L'Ecuyer (1999) ; L'Ecuyer et Touzin (2000).

**Exemple 20** (MRG32k3a).  $J = 2, k = 3$ ,

$$\begin{aligned} m_1 &= 2^{32} - 209, a_{11} = 0, a_{12} = 1403580, a_{13} = -810728, \\ m_2 &= 2^{32} - 22853, a_{21} = 527612, a_{22} = 0, a_{23} = -1370589. \end{aligned}$$

*Combinaison* :  $z_n = (x_{1,n} - x_{2,n}) \pmod{m_1}$ .

Le générateur correspond à un MRG caractérisé par  $k = 3$ ,  $m = m_1 m_2 = 18446645023178547541$ , et les paramètres  $a_1 = 18169668471252892557$ ,  $a_2 = 3186860506199273833$ ,  $a_3 = 8738613264398222622$ . Sa période  $\rho$  vaut  $(m_1^3 - 1)(m_2^3 - 1)/2 \approx 2^{191}$ .

```
#define norm 2.328306549295728e-10 /* 1/(m1+1) */
#define m1 4294967087.0
#define m2 4294944443.0
#define a12 1403580.0
#define a13n 810728.0
#define a21 527612.0
#define a23n 1370589.0

double s10, s11, s12, s20, s21, s22;

double MRG32k3a ()
{
    long k;
    double p1, p2;

    /* Component 1 */
    p1 = a12 * s11 - a13n * s10;
    k = p1 / m1; p1 -= k * m1; if (p1 < 0.0) p1 += m1;
    s10 = s11; s11 = s12; s12 = p1;

    /* Component 2 */
    p2 = a21 * s22 - a23n * s20;
    k = p2 / m2; p2 -= k * m2; if (p2 < 0.0) p2 += m2;
    s20 = s21; s21 = s22; s22 = p2;

    /* Combination */
    if (p1 <= p2) return ((p1 - p2 + m1) * norm);
    else return ((p1 - p2) * norm);
}
```

## 4.6 Générateurs basés sur des récurrences linéaires dans $\mathcal{F}_2$

Considérons le champs de Galois (i.e. un corps ne contenant qu'un nombre fini d'éléments)  $\mathcal{F}_2$ . Il s'agit de l'ensemble  $\{0, 1\}$  sur lequel sont définies les opérations d'addition et de multiplication modulo 2. A partir du vecteur d'état  $\mathbf{x}_{n-1}$ , à l'étape  $n - 1$ , nous définissons comme précédemment la récurrence linéaire :

$$\mathbf{x}_n = X\mathbf{x}_{n-1}.$$

#### 4.6. GÉNÉRATEURS BASÉS SUR DES RÉCURRENCES LINÉAIRES DANS $\mathcal{F}_2$ 57

Toutefois,  $\mathbf{x}_n$  est exprimé à présent comme un vecteurs de  $k$  bits, de sorte que chaque composante se trouve bien dans  $\mathcal{F}_2$ . Le vecteur de sortie  $\mathbf{y}_n$ , défini sur  $w$  bits, est obtenu comme suit :

$$\mathbf{y}_n = B\mathbf{x}_n.$$

Il nous reste à définir la sortie, à savoir un réel compris dans l'intervalle  $[0, 1)$  :

$$u_n = \sum_{j=1}^w y_{n,j-1} 2^{-j} = .y_{n,0} y_{n,1} y_{n,2} \cdots$$

Chaque coordonnée de  $\mathbf{x}_n$  et de  $\mathbf{y}_n$  suit la récurrence linéaire

$$x_{n,j} = (\alpha_1 x_{n-1,j} + \cdots + \alpha_k x_{n-k,j}) \bmod 2,$$

où  $a_1, a_2, \dots, a_k \in \{0, 1\}$ , ce qui permet de définir le

$$P(z) = z^k - \alpha_1 z^{k-1} - \cdots - \alpha_{k-1} z - \alpha_k = \det(X - zI).$$

La période maximale  $\rho = 2^k - 1$  est atteinte ssi  $P(z)$  est primitif sur  $\mathcal{F}_2$  (c'est-à-dire qu'il n'est pas factorisable en produit de polynômes).

De même que pour les MRG's, il est possible de sauter en avant dans la séquence ainsi produite, par blocs de  $i$  étapes, en procédant comme suit :

$$\mathbf{x}_{n+i} = \underbrace{(X^i \bmod 2)}_{\text{précalculer}} \mathbf{x}_n \bmod 2.$$

##### 4.6.1 Générateur de Tausworthe (ou LFSR : linear feedback shift register)

De tels générateurs ont été introduits par Tausworthe, en 1965. Nous définissons une séquence  $x_1, x_2, \dots$  de chiffres binaires par la récurrence

$$x_n = (a_1 x_{n-1} + \cdots + a_k x_{n-k}) \bmod 2, \quad (4.3)$$

où  $a_1, a_2, \dots, a_k \in \{0, 1\}$ , et  $a_k \neq 0$ . Le nom du générateur vient de la possibilité d'utilisation d'un registre de décalage (shift register), alimenté par une fonction de feedback linéaire, comme illustré sur la Figure 4.6.1, avec 16 bits. La fonction de sortie est définie comme

$$u_n = \sum_{l=1}^w x_{nv+l-1} 2^{-l} = .x_{nv} x_{nv+1} x_{nv+2} \cdots x_{nv+w-1}.$$

Cela revient à choisir

$$X = \begin{pmatrix} a_k & a_{k-1} & \dots & a_1 \\ 1 & & & 0 \\ & \ddots & & 0 \\ & & 1 & 0 \end{pmatrix}^\nu \text{ et } B = I.$$

Comme précédemment, la période maximale est  $\rho = 2^k - 1$ , laquelle est atteinte ssi  $Q(z) = z^k - a_1 z^{k-1} - \dots - a_{k-1} z - a_k$  est primitif et le plus grand commun diviseur de  $\nu$  et  $2^k - 1$  vaut 1.

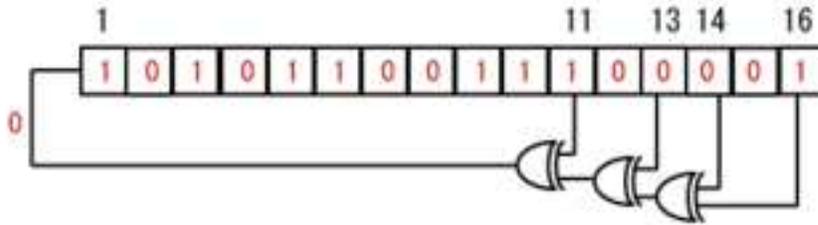


FIGURE 4.8 – Registre de décalage à feedback linéaire.

Dans la plupart des applications, seulement deux des coefficients sont non nuls pour simplifier l'implantation, ce qui donne le trinôme :  $Q(z) = z^k - a_r z^{k-r} - a_k$ . Comme on travaille dans  $\mathcal{F}_2$ , (4.3) devient

$$x_n = (x_{n-r} + x_{n-k}) \pmod{2}.$$

L'exécution de l'addition modulo 2 est équivalente à l'instruction ou-exclusif (xor) sur les bits :

$$x_n = \begin{cases} 0 & \text{si } x_{n-r} = x_{n-k}, \\ 1 & \text{si } x_{n-r} \neq x_{n-k}. \end{cases}$$

Plus généralement, on construit une implantation très rapide par des shifts, xors, masques, etc., si  $\nu \leq r$  et  $2r > k$ .

Les générateurs LFSR sont connus pour avoir des déficiences statistiques. On peut cependant en améliorer les propriétés en considérant des LFSR combinés.

### 4.6.2 Generalized feedback shift register (GFSR)

Introduit par Lewis et Payne [11], ce générateur se base sur la récurrence

$$\mathbf{v}_n = (a_1 \mathbf{v}_{n-1} + \dots + a_r \mathbf{v}_{n-r}) \pmod{2} = (v_{n,0}, \dots, v_{n,w-1})^T,$$

et

$$\mathbf{y}_n = \mathbf{v}_n.$$

Si  $P(z)$  est un trinôme, ce qui est courant dans les implantations, nous avons

$$\mathbf{v}_n = (\mathbf{v}_{n+m-r} + \mathbf{v}_{n-r}) \pmod{2},$$

ce qui donne

$$X = \begin{pmatrix} & & & I_w & & I_w \\ & I_w & & & & \\ & & I_w & & & \\ & & & I_w & & \\ & & & & \ddots & \\ & & & & & I_w \end{pmatrix}$$

Plus généralement,  $P(z) = z^r - a_1 z^{r-1} - \dots - a_{r-1} z - a_r$  et la période maximale est  $2^r - 1$  même si l'état a  $rw$  bits.

Ceci signifie que nous utilisons  $w$  copies de la récurrence (4.3), avec des valeurs initiales différentes, et on utilise une copie pour chaque chiffre de l'expansion fractionnelle de  $u_n$ . Si  $\{x_{j,n}\}$  désigne la  $j^{\text{e}}$  copie et si  $x_{j,n} = x_{n+d_j}$  pour tout  $j, n$ ,

$$u_n = \sum_{j=1}^w x_{n+d_j} 2^{-j}.$$

Souvent,  $d_j = (j-1)d$ , pour un certain  $d$  fixé

### 4.6.3 Twisted GFSR

Ces générateurs, proposés par Matsumoto et Kurita 1992, 1994, généralisent les GFSR comme suit :

$$\begin{aligned} \mathbf{v}_n &= (\mathbf{v}_{n+m-r} + A\mathbf{v}_{n-r}) \pmod{2} \\ \mathbf{y}_n &= \mathbf{v}_n \text{ ou } \mathbf{y}_n = T\mathbf{v}_n, \\ X &= \begin{pmatrix} & & & I_w & A \\ I_w & & & & \\ & I_w & & & \\ & & I_w & & \\ & & & \ddots & \\ & & & & I_w \end{pmatrix}. \end{aligned}$$

La période maximale est  $2^{rw} - 1$ , atteinte ssi  $Q(z^r + z^m)$  est primitif de degré  $rw$ , où  $Q$  est le polynôme caractéristique de  $A$  (i.e.  $\det(XI - A)$ ). L'exemple le plus connu est le générateur TT800, qui atteint une période de  $2^{800} - 1$ .

Récurrance multiple matricielle (Niederreiter 1995) :

$$\begin{aligned} \mathbf{v}_n &= A_1\mathbf{v}_{n-1} + A_2\mathbf{v}_{n-2} + \dots + A_r\mathbf{v}_{n-r}, \\ \mathbf{y}_n &= \mathbf{v}_n, \\ X &= \begin{pmatrix} A_1 & A_2 & \dots & A_{r-1} & A_r \\ I_w & & & & \\ & I_w & & & \\ & & \ddots & & \\ & & & I_w & \end{pmatrix}. \end{aligned}$$

La période maximale est  $2^{rw} - 1$ .

La période peut être considérablement améliorée en considérant une récurrence de la forme

$$\mathbf{v}_n = (\mathbf{v}_{n+m-r} + A(\mathbf{v}_{n-r}^u | \mathbf{v}_{n-r+1}^l)) \pmod{2}, \quad (4.4)$$

$$\mathbf{y}_n = T\mathbf{v}_n. \quad (4.5)$$

L'exposant  $u$  signifie que nous prenons les  $(w-p)$  bits de poids forts, et  $l$ , les  $p$  bits de poids faible. Cette relation est connue sous le nom Mersenne Twister, comme



En combinant des LFSR, TGFSR, ou Mersenne twister entre eux, on obtient des générateurs ayant de bien meilleures équidistributions.

Exemple : LFSR113.

```
unsigned long z1, z2, z3, z4;

double lfsr113 ()
{
    /* Generates numbers between 0 and 1. */
    unsigned long b;
    b = (((z1 << 6) ^ z1) >> 13);
    z1 = (((z1 & 4294967294) << 18) ^ b);
    b = (((z2 << 2) ^ z2) >> 27);
    z2 = (((z2 & 4294967288) << 2) ^ b);
    b = (((z3 << 13) ^ z3) >> 21);
    z3 = (((z3 & 4294967280) << 7) ^ b);
    b = (((z4 << 3) ^ z4) >> 12);
    z4 = (((z4 & 4294967168) << 13) ^ b);
    return ((z1 ^ z2 ^ z3 ^ z4) * 2.3283064365387e-10);
}
```

Les opérations utilisées sont

- & : opérateur et.
- ^ : ou exclusif.
- << : décale à gauche ; revient à multiplier par une puissance de 2.
- >> : décale à droite ; revient à diviser par une puissance de 2.

## 4.8 Tests statistiques

Bien que l'étude des propriétés théoriques d'un générateur permette de déterminer a priori si un tel générateur satisfait certains critères nécessaires pour obtenir de bons résultats, il est ultimement indispensable de recourir à des tests statistiques pour valider celui-ci. Nous formulons l'hypothèse

$$\mathcal{H}_0 : \{u_0, u_1, u_2, \dots\} \text{ sont des variables aléatoires i.i.d. } U(0, 1)''.$$

Nous savons que  $\mathcal{H}_0$  est fautive, mais comment pouvons-nous le détecter ? Pour ce faire, définissons une statistique  $T$ , fonction de  $u_i$ , dont la distribution sous  $\mathcal{H}_0$  est connue (ou approximativement). Nous rejetons  $\mathcal{H}_0$  si la valeur de  $T$  est trop extrême.

```

int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
              // guaranteed to be random.
}

```

FIGURE 4.9 – The XKCD random numbers generator (<http://xkcd.com/c221.html>).

La puissance et l'efficacité du test dépendent fortement de la classe d'alternatives. Différents tests détectent des problèmes différents. L'idéal serait de disposer d'une statistique  $T$  qui imite la variable aléatoire d'intérêt pratique. Ce n'est cependant guère réalisable en pratique. A défaut, nous pourrions chercher à construire un générateur de variables aléatoires qui passerait tous les tests, mais c'est formellement impossible. Il nous faut donc un compromis : construire un générateur qui passe le plus de tests raisonnables. La plupart des tests étudient la répartition uniforme d'un ensemble ou sous-ensemble quelconque de points produits par le générateur. Avoir une bonne couverture uniforme ne suffit toutefois pas : les points générés doivent avoir l'apparence d'indépendance les uns par autres. Les tests d'uniformité, qui représente la majeure partie des tests à notre disposition, remplissent partiellement ce second objectif dans la mesure où ils sont valides pour des sous-ensembles de points très variés. Certains tests cependant sont spécifiquement conçus pour tester les corrélations entre les points produits par un générateur.

Le logiciel TestU01 (L'Ecuyer et Simard), disponible à l'adresse <http://www.iro.umontreal.ca/~simardr/testu01/tu01.html>, implémente un nombre très important de tels tests, et les décrire tous dépasserait largement les objectifs du présent document. Nous nous limiterons à quelques exemples afin d'illustrer les idées sous-jacentes.

#### 4.8.1 Test de collisions

Partitionnons l'hypercube  $[0, 1]^t$  en  $k = d^t$  boîtes cubiques de tailles égales. Générons  $n$  points  $\mathbf{u}_i = (u_{ti}, \dots, u_{ti+t-1})$  dans  $[0, 1]^t$ , et posons  $X_j$  le nombre de points dans la boîte  $j$ . Le nombre de collisions est donné par  $[C = \sum_{j=0}^{k-1} \max(0, X_j -$

1).] Sous  $\mathcal{H}_0$ ,  $C$  suit approximativement une loi de Poisson de moyenne  $\lambda = n^2/(2k)$ , pour  $n$  suffisamment grand, et un petit  $\lambda$ .

Si nous observons  $c$  collisions, nous calculons la  $p$ -valeur à droite comme

$$p^+(c) = P[X \geq c \mid X \sim \text{Poisson}(\lambda)].$$

Nous rejetons  $\mathcal{H}_0$  si  $p^+(c)$  est de manière consistante très proche de 0 (trop de collisions) ou de 1 (pas assez de collisions).

Une interprétation possible est l'étude par simulation Monte Carlo du comportement probabiliste d'un algorithme de hachage.

### 4.8.2 Espacement entre anniversaires

Nous partitionnons à nouveau l'hypercube  $[0, 1]^t$  en  $k = d^t$  boîtes cubiques, et générons  $n$  points. Soit  $I_{(1)} \leq I_{(2)} \leq \dots \leq I_{(n)}$  les numéros de boîte où les points tombent, triés par ordre croissant. Nous calculons les espacements  $S_j = I_{(j+1)} - I_{(j)}$ ,  $1 \leq j \leq n - 1$ . Le nombre de collisions entre les espacements est défini comme  $[Y = \sum_{j=1}^{n-1} I[S_{(j+1)} = S_{(j)}].$  Pour  $k$  suffisamment grand,  $Y$  suit approximativement une loi de Poisson de moyenne  $\lambda = n^3/(4k)$ . Si  $Y$  prend la valeur  $y$ , la  $p$ -valeur à droite est

$$p^+(y) = P[X \geq y \mid X \sim \text{Poisson}(\lambda)].$$

Interprétation comme problème de simulation : nous pouvons vouloir étudier la distribution de  $Y$  par simulation Monte Carlo, pour des valeurs modérées de  $k$  et  $n$ .

### 4.8.3 Test d'autocorrélation

Générons  $n$  uniformes  $u_1, \dots, u_n$  et calcule l'autocorrélation empirique de distance  $k$  :

$$\hat{\rho}_k = \frac{1}{n-k} \sum_{j=1}^{n-k} \left( u_j u_{j+k} - \frac{1}{4} \right).$$

La distribution empirique des  $N$  valeurs de  $\sqrt{12(n-k)}\hat{\rho}_k$  est comparée avec la distribution normale standard, qui est la distribution théorique asymptotique quand

$n \rightarrow +\infty$ . L'approximation n'est valide que pour  $n$  très grand, et avec  $k$  très petit en comparaison de  $n$ .

D'autres tests tiennent compte de la dépendance créée entre des vecteurs se recouvrant partiellement. Pour plus d'informations, le lecteur est renvoyé à la documentation de `TestU01`.

## 4.9 Quelques générateurs largement utilisés

Avant 2003, le générateur dans MS Excel était

$$u_i = (9821.0 u_{i-1} + 0.211327) \bmod 1.$$

Le générateur est à présent une combinaison de trois GCLs simplistes. A la même période, le générateur dans MS VisualBasic était

$$\begin{aligned} x_i &= (1140671485 x_{i-1} + 12820163) \bmod 2^{24}, \\ u_i &= x_i / 2^{24}. \end{aligned}$$

Le générateur dans `java.util.Random` de Java est quant à lui

$$\begin{aligned} x_{i+1} &= (25214903917 x_i + 11) \bmod 2^{48} \\ u_i &= [2^{27} (x_{2i} \bmod 2^{26}) + x_{2i+1} \bmod 2^{27}] / 2^{53} \end{aligned}$$

## 4.10 Résultats de tests

### 4.10.1 Test de colisions

Considérons  $t = 2$ ,  $d = n/16$ ,  $\lambda = 128$ .

$n$	Java		VisualBasic		Excel	
	c	$p^+(c)$	c	$p^+(c)$	c	$p^+(c)$
$2^{14}$					132	
$2^{15}$			75	$1 - 3.1 \times 10^{-7}$	128	
$2^{16}$			38	$> 1 - 10^{-15}$	121	
$2^{17}$			0	$> 1 - 10^{-15}$	170	$2.2 \times 10^{-4}$
$2^{18}$			0	$> 1 - 10^{-15}$	202	$9.5 \times 10^{-10}$
$2^{19}$			0	$> 1 - 10^{-15}$	429	$< 10^{-15}$

Si on rejette les dix premiers bits de chaque  $u_i$  : on prend  $u_i := 1024u_i \bmod 1$ .

$n$	Java		VisualBasic		Excel	
	$c$	$p^+(c)$	$c$	$p^+(c)$	$c$	$p^+(c)$
$2^{14}$			8192	$< 10^{-15}$		
$2^{15}$			24576	$< 10^{-15}$		
$2^{16}$			57344	$< 10^{-15}$		
$2^{17}$			122880	$< 10^{-15}$		
$2^{18}$			253952	$< 10^{-15}$	224	$1.0 \times 10^{-14}$
$2^{19}$	160	$3.5 \times 10^{-3}$	516096	$< 10^{-15}$	425	$< 10^{-15}$

#### 4.10.2 Espacements d'anniversaire

Considérons  $t = 2, \lambda = 1$ .

$n$	$d$	Java		VisualBasic		Excel	
		$y$	$p^+(y)$	$y$	$p^+(y)$	$y$	$p^+(y)$
$2^8$	$2^{11}$						
$2^{10}$	$2^{14}$			10	$1.1 \times 10^{-7}$		
$2^{12}$	$2^{17}$			592	$< 10^{-15}$	5	$3.7 \times 10^{-3}$
$2^{14}$	$2^{20}$			11129	$< 10^{-15}$	71	$< 10^{-15}$
$2^{16}$	$2^{23}$			64063	$< 10^{-15}$	558	$< 10^{-15}$
$2^{18}$	$2^{26}$	14	$4.5 \times 10^{-12}$	261604	$< 10^{-15}$	4432	$< 10^{-15}$

$n$	$d$	LCG-16807	
		$y$	$p^+(y)$
$2^{10}$	$2^{14}$		
$2^{12}$	$2^{17}$	2	
$2^{14}$	$2^{20}$	170	$< 10^{-15}$
$2^{16}$	$2^{23}$	10060	$< 10^{-15}$

Et en rejetant les 10 premiers bits :

#### 4.11. TESTS SYSTÉMATIQUES POUR DES FAMILLES DE GÉNÉRATEURS DE NOMBRES ALÉATOIRES 67

$n$	Java		VisualBasic		Excel	
	$y$	$p^+(y)$	$y$	$p^+(y)$	$y$	$p^+(y)$
$2^8$			52	$< 10^{-15}$		
$2^9$			199	$< 10^{-15}$		
$2^{10}$			672	$< 10^{-15}$		
$2^{11}$			1754	$< 10^{-15}$		
$2^{12}$			3901	$< 10^{-15}$		
$2^{13}$			8102	$< 10^{-15}$		
$2^{14}$			16374	$< 10^{-15}$		
$2^{15}$	21	$6.0 \times 10^{-15}$	32763	$< 10^{-15}$		
$2^{16}$	99	$< 10^{-15}$	65531	$< 10^{-15}$	7	$4.5 \times 10^{-3}$
$2^{17}$	697	$< 10^{-15}$	—	—	34	$< 10^{-15}$
$2^{18}$	639	$< 10^{-15}$	—	—	186	$< 10^{-15}$

### 4.11 Tests systématiques pour des familles de générateurs de nombres aléatoires

Pour une famille de générateurs de nombres aléatoires, on cherche une relation du type  $[n_0 \approx K\rho^\gamma]$ , où  $n_0$  est la taille d'échantillon minimum pour obtenir un fort rejet,  $\rho$  est la longueur de période, et  $K$  et  $\gamma$  sont des constantes.

Pour des GCLs, on obtient (grossièrement) pour le test de collisions :

$$n_0 \approx 16\rho^{1/2},$$

et pour le test d'espacement entre anniversaires :

$$n_0 \approx 16\rho^{1/3}.$$

Nous souhaitons dès lors construire le générateur de nombres aléatoires avec  $\rho$  suffisamment grand de sorte que générer  $n_0$  nombres est infaisable en pratique.

#### 4.11.1 Librairie de tests

Tests effectués sur la plupart des générateurs existants. Pratiquement aucun des générateurs présent dans les logiciels commerciaux ne passe tous les tests. Vrai en

particulier pour les générateurs modulo-2.

Les vainqueurs sont des MRG avec une bonne période et une bonne structure, des générateurs multiplicatifs “lagged-Fibonacci”, des générateurs non-linéaires conçus pour la cryptologie, et certains générateurs avec des composantes venant de différentes familles.

Il est possible de combiner efficacité empirique et théorique, comme l’illustre le MRG32k3a.

Note : générateur multiplicatifs “lagged-Fibonacci”. Forme de base

$$x_n = x_{n-r} * x_{n-k} \text{ mod } m.$$

Il a moins de propriétés mathématiques connues, difficultés du choix de  $r$  et de  $k$ , initialisation complexe.  $m$  est souvent pris comme une puissance de 2.

## 4.12 Exercices

1. Prouvez que si  $X = F^{-1}(U)$ , où  $U \sim \text{Uniform}(F(a), F(b))$ , alors  $X$  a la distribution  $F$  tronquée sur l’intervalle  $(a, b)$ .  
Si  $F$  est continue, cela revient à tronquer sur l’intervalle  $[a, b]$ , mais pas si  $F$  présente un saut en  $a$ . Pourquoi ? Comment généreriez-vous  $X$  à partir de la distribution  $F$  tronquée sur  $[a, b]$  dans ce dernier cas ?

## Chapitre 5

# Génération de variables aléatoires non uniformes

### 5.1 Motivations

Supposons que nous disposons d'un bon générateur fournissant des variables aléatoires i.i.d.  $U(0, 1)$ , comme décrit au chapitre précédent. En pratique, nous souhaitons cependant générer des variables aléatoires plutôt selon des lois diverses (normale, Weibull, Poisson, binomiale, etc.), voire d'autres "objets" aléatoires : processus stochastiques, points sur une sphère, matrices aléatoires, arbres aléatoires, etc. Pour ce faire, il convient donc de pouvoir transformer une uniforme  $U(0, 1)$  de manière adéquate.

Les propriétés recherchées pour une méthode de transformation sont :

- méthode correcte (ou très bonne approximation) ;
- simple : facile à comprendre et à implanter. Un algorithme plus efficace et plus complexe peut être approprié pour un usage précis dans un logiciel permanent, mais moins pour un modèle de simulation utilisé une seule fois ;
- rapide : après un temps d'initialisation ("setup"), si requis, le temps marginal par appel devrait être aussi faible que possible. Il faut parfois faire un compromis entre les deux : dans la plupart des simulations, nous générerons un grand nombre de variables aléatoires, aussi le temps marginal est habituellement plus important que le temps s'initialisation. Mais si les paramètres de la simulation change souvent ou de manière aléatoire au cours de la simulation, le temps d'initialisation peut devenir un point important à considérer ;

- mémoire utilisée : on peut souvent améliorer la vitesse au prix du stockage de tableaux de tailles importantes ;
- robustesse : l'algorithme doit être précis et efficace pour toutes les valeurs des paramètres qui peuvent nous intéresser.

Nous privilégierons aussi les approches compatibles avec les méthodes de réduction de variance (Chapitre 7). Par exemple, nous préférons habituellement l'inversion parce que cela facilite la synchronisation lorsque l'on compare deux systèmes ou lorsqu'on veut utiliser des variables de contrôle, ou des valeurs antithétiques, etc. Nous serons prêts à sacrifier un peu la vitesse pour préserver l'inversion. Les méthodes de réduction de la variance nous feront au final souvent gagner beaucoup plus de temps de calcul que ce que l'on aura sacrifié pour préserver l'inversion.

## 5.2 Inversion

La technique majeure pour générer une v.a.  $X$  est l'inversion. Considérons la fonction de répartition  $F$  de  $X$ . Soit  $U \sim U(0, 1)$  et

$$X = F^{-1}(U) = \min\{x : F(x) \geq U\}.$$

Alors

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

i.e.,  $X$  a la fonction de répartition voulue. Dans le cas continu,  $F(X) \sim U[0, 1]$ , aussi la justesse de l'approche est immédiate. Dans le cas discret, il s'agit de prouver que  $P[X = x_i] = p(x_i)$ , pour tout  $i$ , et on supposera  $x_1 < x_2 < \dots < x_n$ . Pour  $i = 1$ , nous avons  $X = x_1$  si et seulement si  $U \leq F(x_1) = p(x_1)$ , comme voulu. Pour  $i \geq 2$ , nous avons  $X = x_i$  si et seulement si  $F_{i-1} < U \leq F(x_i)$ . Comme  $0 \leq F(x_{i-1}) < F(x_i) \leq 1$ , nous obtenons

$$P[X = x_i] = P[F(x_{i-1}) < U \leq F(x_i)] = F(x_i) - F(x_{i-1}) = p(x_i).$$

Il est facile de prouver que le principe fonctionne toujours pour des distributions mixtes, c'est-à-dire ayant des composantes discrètes et continues.

La technique d'inversion a pour avantage d'être monotone : il y a un seul  $U$  pour chaque  $X$ . Mais pour certaines lois,  $F$  peut être très difficile à inverser. Toutefois, nous pouvons souvent quand même approximer  $F^{-1}$  numériquement. De manière générale, on peut utiliser des techniques d'interpolation (par exemple une interpolation de l'Hermite ou, dans le cadre de l'estimation de modèles, à l'aide de B-splines).

**Exemple 21** (Loi triangulaire). *Pour générer une triangulaire définie sur l'intervalle  $[a, b]$  et de mode  $c$ , par inversion, on commencera par tirer  $u$  suivant une uniforme  $U(0, 1)$ . Il s'agit tout d'abord de déterminer si nous sommes à gauche ou à droite du mode. Partant de la fonction de densité*

$$f(x) = \begin{cases} \frac{2}{b-a} \frac{x-a}{c-a} & \text{if } a \leq x \leq c, \\ \frac{2}{b-a} \frac{b-x}{b-c} & \text{if } c \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases}$$

*Nous avons que la probabilité d'obtenir un nombre inférieur à  $c$  est*

$$\frac{2}{b-a} \int_a^c \frac{x-a}{c-a} = \frac{x^2 - 2ax}{(b-a)(c-a)} \Big|_a^c = \frac{c-a}{b-a}.$$

*Comme la fonction de répartition est*

$$\begin{cases} \frac{(x-a)^2}{(b-a)(c-a)} & \text{if } a \leq x \leq c, \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & \text{if } c \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases}$$

*il suffit d'inverser cette fonction et par conséquent, de retourner*

$$\begin{cases} a + \sqrt{(b-a)(c-a)u} & \text{si } u \leq \frac{c-a}{b-a}, \\ b - \sqrt{(b-a)(b-c)(1-u)} & \text{sinon.} \end{cases}$$

**Exemple 22** (loi normale). *Si  $Z \sim N(0, 1)$ , alors  $X = \sigma Z + \mu : N(\mu, \sigma^2)$ . Il suffit donc de savoir générer une  $N(0, 1)$ , de densité :  $f(x) = (2\pi)^{-1/2} e^{-x^2/2}$ . Mais nous n'avons pas de formule pour  $F(x)$  ni pour  $F^{-1}(x)$ . Par contre, on sait que quand  $x$  est grand,  $F(x)$  ressemble à*

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

*dont l'inverse<sup>1</sup> est  $\tilde{F}^{-1}(u) = \sqrt{-2 \ln \sqrt{2\pi}(1-u)}$ .*

*L'idée de base est, pour  $x > 0$ , d'approximer  $x = F^{-1}(u)$  par  $y = \tilde{F}^{-1}(u)$ , plus un quotient de 2 polynômes en  $y$  (i.e., une fonction rationnelle) qui approxime la différence. Le cas  $x < 0$  en découle directement en raison de la symétrie de  $F^{-1}(\cdot)$  autour de 0.5.*

---

1. Nous négligeons le terme  $-\ln x$  comme  $x$  est grand, de sorte que l'expression est dominée par  $-x^2/2$ .

Cela donne, si  $U > 1/2$ ,

$$Y = \sqrt{-2\ln[(1-U)\sqrt{2\pi}]},$$

$$X = Y + \frac{p_0 + p_1Y + p_2Y^2 + p_3Y^3 + p_4Y^4}{q_0 + q_1Y + q_2Y^2 + q_3Y^3 + q_4Y^4}$$

Les  $p_i$  et  $q_i$  sont choisis de manière à ce que l'approximation soit excellente pour tout  $U > 1/2$ . Si  $U < 1/2$ , on utilise la symétrie : on calcule  $X$  pour  $1 - U$  au lieu de  $U$ , puis on retourne  $-X$ .

Wichura [19] a ainsi proposé un algorithme basé sur ce principe qui permet de calculer la fonction de répartition inverse jusqu'aux environs de 16 chiffres significatif, soit de l'ordre de la précision machine, pour autant  $\min\{u, 1 - u\}$  soit plus grand que  $10^{-316}$ .

Pour des distributions telles que la  $\chi^2$ , la gamma, la beta, etc., les choses se compliquent car la forme de  $F^{-1}$  dépend des paramètres.

### 5.2.1 Implantation de l'inversion pour les lois discrètes

Rappelons que

$$p(x_i) = P[X = x_i];$$

$$F(x) = \sum_{x_i \leq x} p(x_i).$$

La technique d'inversion revient dès lors à générer  $U$ , trouver  $I = \min\{i | F(x_i) \geq U\}$  et retourner  $x_I$ .

Les algorithmes suivants font cela.

**Algorithm 5.1:** Initialisation : mettre les  $x_i$  et les  $F(x_i)$  dans des tableaux, pour  $i = 1, \dots, n$ .

1. Recherche linéaire (temps en  $O(n)$ ) :  
 $U \leftarrow U(0, 1)$ ;  $i \leftarrow 1$ ; tant que  $F(x_i) < U$  faire  $i \leftarrow i + 1$ ;  
 retourner  $x_i$ .
2. Recherche binaire (temps en  $O(\log(n))$ ) :  
 $U \leftarrow U(0, 1)$ ;  $L \leftarrow 0$ ;  $R \leftarrow n$ ;  
 tant que  $L < R - 1$

$m \leftarrow \lfloor (L + R)/2 \rfloor$ ;  
 si  $F(x_m) < U$  alors  $L \leftarrow m$  sinon  $R \leftarrow m$  ;  
 (\* Invariant : l'indice  $I$  est dans  $\{L + 1, \dots, R\}$ . \*)  
 retourner  $x_R$ .

## 5.2.2 Inversion pour l'interpolation linéaire

Utile, par exemple, pour générer des v.a. selon une interpolation linéaire par morceaux de  $\hat{F}$ . Supposons que  $F$  est linéaire par morceaux, avec des cassures à  $x_0 < x_1 < \dots < x_n$ . Pour un  $U$  donné, pour trouver  $F^{-1}(U)$ , on doit d'abord trouver le bon morceau linéaire, c'est-à-dire l'indice  $i$  tel que

$$F(x_{i-1}) < U \leq F(x_i).$$

Pour cela, on peut utiliser une recherche binaire. Puis on retourne  $X$  tel que

$$\frac{U - F(x_{i-1})}{F(x_i) - F(x_{i-1})} = \frac{X - x_{i-1}}{x_i - x_{i-1}},$$

i.e.,

$$\begin{aligned}
 X &= x_{i-1} + \frac{(U - F(x_{i-1}))(x_i - x_{i-1})}{F(x_i) - F(x_{i-1})} \\
 &= \frac{(F(x_i) - U)x_{i-1} + (U - F(x_{i-1}))x_i}{F(x_i) - F(x_{i-1})} \\
 &= A_i + B_i U.
 \end{aligned}$$

## 5.2.3 Processus de Poisson stationnaire

Pour un processus de Poisson  $\{N(t), t \geq 0\}$ , on sait que

$$\tilde{N}(x) = N(\Lambda^{-1}(x)),$$

où  $\{\tilde{N}(x), x \geq 0\}$  est un processus de Poisson standard, i.e. processus stationnaire de taux 1 (voir Section A.11.3).

Rappel : fonction de taux cumulé  $\Lambda(t) = E[N(t)]$ , où  $N(t)$  est le nombre d'arrivées durant  $[0, t]$ .

On simule les instants d'arrivées  $X_0, X_1, X_2, \dots$  d'un processus de Poisson standard, puis on calcule les instants d'arrivées du processus original via  $T_i = \Lambda^{-1}(X_i)$ .

Si la fonction de taux  $\lambda(t)$  est constante par morceaux, alors  $\Lambda(t)$  est linéaire par morceaux et on peut l'inverser comme lorsque  $F$  est linéaire par morceaux.

En pratique, si  $\lambda(t)$  ne change pas trop souvent, on va générer  $X_{i+1} - X_i \sim \text{Exponential}(1)$ , poser temporairement

$$T_{i+1} = T_i + (X_{i+1} - X_i)/\lambda(T_i),$$

puis réévaluer  $T_{i+1}$  au temps  $\tilde{t}$  si jamais le taux  $\lambda(t)$  change au temps  $\tilde{t}$  pour  $T_i < \tilde{t} < T_{i+1}$ .

## 5.3 Autres approches

### 5.3.1 Composition

Supposons que  $F$  est une combinaison convexe de plusieurs fonctions de répartition :

$$F(x) = \sum_{j=0}^{\infty} p_j F_j(x),$$

avec  $\sum_{j=0}^{\infty} p_j = 1$ , et qu'il est plus facile d'inverser  $F_j$ ,  $j = 0, \dots, \infty$  que  $F$ . Un algorithme simple pour tirer de  $F$  dans pareil cas est décrit ci-dessous.

**Algorithm 5.2:** Génération par composition      Générer  $J = j$  avec probabilité  $p_j$ , puis générer  $X$  selon  $F_j$ .

La méthode requiert dès lors deux uniformes pour chaque variable, et exploite la décomposition

$$P[X \leq x] = \sum_{j=0}^{\infty} P[X \leq x | J = j] P[J = j] = \sum_{j=0}^{\infty} F_j(x) p_j,$$

qui montre qu'il suffit de générer  $X$  en conditionnant sur la valeur de  $J$ .

### 5.3.2 Convolution

Supposons que la variable aléatoire  $X$  qui nous intéresse puisse s'écrire comme suit :

$$X = Y_1 + Y_2 + \dots + Y_n,$$

où les  $Y_i$  sont indépendantes, de lois spécifiées. Il suffit alors de générer les  $Y_i$  et de sommer.

Exemples : Erlang (somme d'exponentielles de même moyenne), binomiale.

## 5.4 Acceptation-rejet

Il s'agit de la technique la plus importante après l'inversion, et est parfois (mais pas toujours !) compatible avec l'utilisation de techniques de réduction de variance. Contrairement à la technique d'inversion, elle ne requiert pas de pouvoir facilement calculer la fonction de répartition, et encore moins son inverse.

Nous allons considérer le cas où  $X$  est continu (le cas discret est analogue). Soit  $f(x)$  la densité de  $X$ , et soit  $t$  une fonction "chapeau" qui majore  $f$ , i.e.  $f(x) \leq t(x)$  pour tout  $x$ . On peut normaliser  $t$  en une densité  $r$  (pour laquelle l'intégration donne 1) :

$$r(x) = t(x)/a, \text{ où } a = \int_{-\infty}^{\infty} t(s)ds.$$

On choisit  $t$  de manière à ce que

- (i) ce soit facile de générer des v.a. de densité  $r$ ,
- (ii)  $a$  soit petit (proche de 1), ou en d'autres termes  $t(x)$  est proche de  $f(x)$ .

Le choix de  $t$  peut être automatisé (voir par exemple Botts [?]).

**Algorithm 5.3:** Méthode d'acceptation-rejet

Répéter  
 Générer  $Y$  de densité  $r(x)$  ;  
 Générer  $U : U(0, 1)$  indépendante de  $Y$  ;  
 jusqu'à obtenir  $U \leq f(Y)/t(Y)$  ;  
 retourner  $Y$ .

**Proposition 2.** La v.a.  $Y$  retournée est de densité  $f$ .

*Démonstration.* Nous avons tout d'abord, comme  $P[U \leq x] = x$ ,

$$\begin{aligned} P[Y \text{ est accepté}] &= P[U \leq f(Y)/t(Y)] \\ &= \int_{-\infty}^{\infty} P[U \leq f(y)/t(y)]r(y)dy \\ &= \int_{-\infty}^{\infty} f(y)/t(y)r(y)dy \\ &= \frac{1}{a}. \end{aligned}$$

Similairement,

$$\begin{aligned} P[Y \leq x \text{ et } Y \text{ est acceptée}] &= \int_{-\infty}^{\infty} P[Y \leq x \text{ et } U \leq f(Y)/t(Y) | Y = y]r(y)dy \\ &= \int_{-\infty}^{\infty} P[y \leq x \text{ et } U \leq f(y)/t(y)]r(y)dy \\ &= \int_{-\infty}^x P[U \leq f(y)/t(y)]r(y)dy \\ &= \int_{-\infty}^x [f(y)/t(y)]r(y)dy \\ &= F(x)/a \end{aligned}$$

Nous pouvons dès lors écrire

$$\begin{aligned} P[Y \leq x | Y \text{ est accepté}] &= \frac{P[Y \leq x \text{ et } Y \text{ est accepté}]}{P[Y \text{ est accepté}]} \\ &= \frac{F(x)/a}{1/a} = F(x). \end{aligned}$$

□

A chaque tour de boucle, la probabilité d'accepter  $Y$  est  $1/a$ . Le nombre de tours de boucle avant l'acceptation est une v.a. géométrique de paramètre  $p = 1/a$ . Le nombre moyen de tours de boucle par v.a. est donc  $a$ .

**Exemple 23.** (Law [8]) Supposons que nous souhaitons tirer des valeurs d'une loi beta de paramètres (4,3), de fonction de densité

$$f(x) = 60x^3(1-x)^2, \text{ pour } 0 < x < 1.$$

$F(x)$  est difficile à inverser, mais il est facile d'établir que le maximum de  $f$  est  $f(0.6) = 2.0736$ . On peut donc prendre  $t(x) = 2.0736$  pour  $0 < x < 1$ . Dans ce

cas  $r(x)$  est la densité  $U(0, 1)$ . On peut diminuer  $a$  en choisissant une densité  $t$  un peu moins simpliste.

**Exemple 24.** Il est également possible de générer une loi  $N(0, 1)$  à partir d'une distribution de Laplace de paramètres  $(0, 1)$ , de densité

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

La fonction de densité de la normale  $N(0, 1)$  étant

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

nous obtenons que

$$t(x) = ar(x) \geq f(x)$$

si et seulement si

$$a \geq \sqrt{\frac{2}{\pi}}e^{-\frac{x^2}{2}+|x|}.$$

Il est facile de montrer que le terme de droite est maximum en  $x = -1$  ou  $x = 1$ , donnant la condition

$$a \geq \sqrt{\frac{2e}{\pi}} \approx 1.3155.$$

En passant à l'égalité, la majoration peut être illustrée comme sur la Figure 5.1, et le rapport  $f(x)/t(x)$  devient

$$\frac{f(x)}{t(x)} = \begin{cases} e^{-\frac{(x-1)^2}{2}} & \text{si } x \geq 0, \\ e^{-\frac{(x+1)^2}{2}} & \text{si } x \leq 0. \end{cases}$$

**Exemple 25.** Technique de découpage ("thinning") pour un processus de Poisson.

Il s'agit d'une variante de la technique d'acceptation/rejet.

Supposons qu'on a un processus de Poisson non stationnaire avec fonction de taux cumulé  $\lambda(t)$  très compliquée (le taux cumulé  $\Lambda(t)$  est difficile à inverser).

Soit  $\bar{\lambda}$  tel que  $\lambda(t) \leq \bar{\lambda}$  pour tout  $t$ .

Algorithme :

Générer des pseudo-arrivées selon un proc. de Poisson de taux constant  $\bar{\lambda}$ .

S'il y a une pseudo-arrivée au temps  $t$ , on l'accepte avec probabilité  $\lambda(t)/\bar{\lambda}$  (elle devient une vraie arrivée), sinon on la rejette.

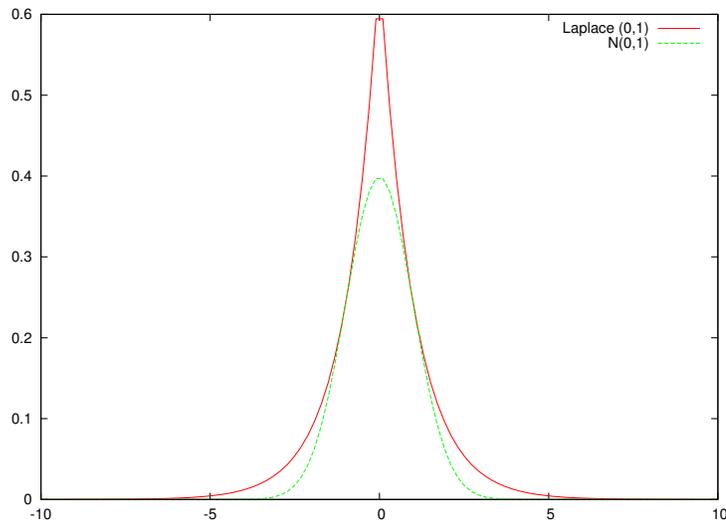


FIGURE 5.1 – Majoration de la Gaussienne par une Laplace (0,1) mise à l'échelle.

## 5.5 Cas particuliers

Bien que les méthodes décrites dans les sections précédentes décrivent les techniques majeures pour générer des variables aléatoires multivariées, il existe de nombreuses approches spécifiques aux distributions considérées. Nous reprenons ici quelques exemples illustratifs, un traitement complet sortant du cadre du présent cours. Ces techniques sont cependant rarement compatibles avec les techniques d'amélioration d'efficacité, et dès lors devraient être employées parcimonieusement. Le lecteur intéressé pourra se référer au livre de Luc Devroye [4], *Non-Uniform Random Variate Generation*, disponible à l'adresse <http://cg.scs.carleton.ca/~luc/rnbookindex.html>, pour une couverture plus exhaustive de la question.

### 5.5.1 Méthode de Box-Muller pour la loi normale

L'exemple le plus connu de traitement particulier concerne la génération de variables aléatoires normales en suivant une transformée de coordonnées cartésiennes en coordonnées polaires. La méthode, proposée en 1958 par Box et Muller [2] a pour avantage d'être exacte, au contraire des approximations faites dans le cadre de

l'inversion, mais numériquement, est plus lente que l'inversion. Son intérêt réside donc d'avantage au niveau théorique qu'au niveau pratique, mais ce type de transformation peut se révéler utile dans d'autres cadres. Considérons un couple  $(X, Y)$  de variables aléatoires  $N(0, 1)$  indépendantes. Leur densité jointe sur  $\mathcal{R}^2$  est

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

Il est aisé de changer les coordonnées cartésiennes  $(X, Y)$  par les coordonnées polaires  $(R, \Theta)$  :

$$R^2 = X^2 + Y^2; Y = R \sin \Theta.$$

La transformée inverse est donnée par  $X = R \cos \Theta$  et  $Y = R \sin \Theta$ . La matrice jacobienne de la transformation  $(r, \theta)$  vers  $(x, y)$  est

$$J = \begin{pmatrix} \cos \theta & -R \sin \theta \\ \sin \theta & R \cos \theta \end{pmatrix},$$

de déterminant  $(\cos \theta)(r \cos \theta) - (\sin \theta)(-r \sin \theta) = r(\cos^2 \theta + \sin^2 \theta) = r$ . Par conséquent,  $(R, \Theta)$  a la densité

$$t(r, \theta) = |J|f(x, y) = r f(r \cos \theta, r \sin \theta) = (r/2\pi)e^{-r^2/2}.$$

Puisque cette densité ne dépend pas de  $\theta$ ,  $\Theta$  doit avoir la densité uniforme sur  $[0, 2\pi]$ . En intégrant par rapport à  $\theta$  sur l'intervalle  $[0, 2\pi]$ , nous obtenons que  $R$  a la densité

$$f_R(r) = \int_0^{2\pi} t(r, \theta) d\theta = r e^{-r^2/2} \quad \text{pour } r \geq 0.$$

La fonction de répartition correspondante est  $F_R(r) = 1 - e^{-r^2/2}$ , et sont inverse est donnée par

$$F_R^{-1}(U) = \sqrt{-2 \ln(1 - U)}.$$

Il suffit alors de générer  $\theta$  et  $R$  indépendamment, en utilisant  $U_1$  et  $U_2$ , puis à transformer ces coordonnées polaires en coordonnées rectangulaires  $(X, Y)$ , autrement dit

$$\begin{aligned} X &= R \cos \theta = \cos(2\pi U_1) \sqrt{-2 \ln(U_2)} \\ Y &= R \sin \theta = \sin(2\pi U_1) \sqrt{-2 \ln(U_2)}. \end{aligned}$$

La transformation à partir de coordonnées cartésiennes vers des coordonnées polaires est également utile pour développer des méthodes élégantes de génération de variables aléatoires pour plusieurs autres distributions (voir Devroye [4]).

### 5.5.2 Variable aléatoire de Poisson

Considérons une variable aléatoire  $X$ , distribuée suivant une loi de Poisson de paramètre  $\lambda$ , avec  $\lambda$  petit. L'idée consiste à simuler un processus de Poisson de taux  $\lambda$  et compter le nombre d'arrivées durant l'intervalle  $(0, 1]$  (cf A.11.3). Nous retournons ensuite ce nombre d'arrivées comme valeur pour  $X$ . Soient  $A_1, A_2, \dots$  les durées entre les arrivées. Ce sont des exponentielles de moyenne  $1/\lambda$ , aussi

$$\sum_{i=1}^X A_i \leq 1 < \sum_{i=1}^{X+1} A_i.$$

Nous pouvons générer chaque  $A_i$  par  $A_i = -\ln(U_i)/\lambda$ . Alors

$$-\sum_{i=1}^X \ln(U_i) \leq \lambda < -\sum_{i=1}^{X+1} \ln(U_i).$$

i.e.

$$\prod_{i=1}^X U_i \geq e^{-\lambda} > \prod_{i=1}^{X+1} U_i,$$

ou

$$X = \max \left\{ k \mid e^{\lambda} \prod_{i=1}^k U_i \geq 1 \right\}.$$

**Algorithm 5.4:** Génération d'une variable aléatoire de Poisson

1. Initialisation :  $X \leftarrow -1; p \leftarrow e^{\lambda};$
2. Répéter  
Générer  $U : U(0, 1); \quad p \leftarrow p \cdot U; \quad X \leftarrow X + 1$   
jusqu'à obtenir  $p < 1;$
3. retourner  $X$ .

### 5.5.3 Uniforme sur la sphère unité

Nous souhaitons générer un vecteur  $U$  uniformément distribué sur  $\mathcal{C}^k = \{x \in \mathcal{R}^k \mid \|x\|_2 = 1\}$ . Ce genre de variables apparaît dans la génération de variables

multivariées elliptiques, utilisées notamment en gestions de risques. La génération de fait en 3 lignes de codes !

```
void generate_hypersphere(Random *ran, int k, double *x)
{
    int i;
    double norm;

    for (i = 0; i < k; i++)
        x[i] = ran_normal_icdf(ran, 0, 1);
    norm = cblas_dnorm2(n, x, 1);
    cblas_dscal(n, 1/norm, x, 1);
}
```

**Proposition 3.** (Devroye, Chap. 5, Section 4)

Si  $N_1, \dots, N_k$  sont des normales i.i.d., alors, en notant  $N = (N_1, \dots, N_k)$ ,  $U = N/\|N\|_2$  est uniformément distribué sur  $\mathcal{C}^k$ .

On peut aussi montrer que  $\|N\|_2$  est indépendant de  $U$ . Dès lors, en utilisant la formule de la variance du produit de deux variables indépendantes, et sachant que  $E[U] = E[N] = 0$ , on obtient que  $Cov[U] = I/Var[\|N\|_2] = I/E[\|N\|_2^2] = I/k$ .

Si  $k = 2$ , nous pouvons calculer un tel point en définissant une variable aléatoire  $\Theta$  uniformément distribuée sur  $[0, 2\pi)$ . Toutefois, le raisonnement ne peut pas s'étendre pour  $k \geq 3$ . Considérons  $k = 3$ . Nous serions alors tentés d'utiliser deux variables aléatoires uniformément distribuées sur  $[0, 2\pi)$ . Notons ces variables  $\Theta$  et  $B$ , et supposons les indépendantes. Nous pouvons faire correspondre à chaque intervalle de  $[0, 2\pi)$  une probabilité, dont la valeur est fixée par la différence entre les extrémités de l'intervalle. Considérons la surface décrite en laissant varier la première variable aléatoire  $\Theta$  de  $\theta_1$  à  $\theta_2$ , et la seconde,  $B$ , de  $\beta_1$  à  $\beta_2$ . La surface vaut, en prenant  $R$  comme rayon de la sphère,

$$\begin{aligned} S &= \left| \int_{\beta_1}^{\beta_2} \int_{\alpha_1}^{\alpha_2} R^2 \cos \beta d\alpha d\beta \right| \\ &= \left| R^2 (\alpha_2 - \alpha_1) (\sin \beta_1 - \sin \beta_2) \right|. \end{aligned}$$

Par analogie avec la cartographie, nous dénommerons  $\alpha$  la longitude et  $\beta$  la latitude. Si la longueur de l'arc décrit en fonction de la latitude ne dépend que de la différence entre les latitudes extrême, il n'en va pas de même pour la longitude : l'arc décrit est d'autant plus petit que la latitude est importante. Si les

points étaient uniformément distribués sur la sphère, à deux surfaces de même valeur devraient correspondre à des probabilités identiques. Nous pouvons constater que suivant la latitude, nous devons varier la différence dans les longitudes. De manière équivalente, nous pouvons dire que les surfaces ne dépendent pas uniquement des différences d'angles. Par conséquent, les points ne sont pas distribués uniformément sur la sphère.

## 5.6 Exercices

1. Prouvez que si  $X = F^{-1}(U)$ , où  $U \sim \text{Uniform}(F(a), F(b))$ , alors  $X$  a la distribution  $F$  tronquée sur l'intervalle  $(a, b]$ .  
Si  $F$  est continue, cela revient à tronquer sur l'intervalle  $[a, b]$ , mais pas si  $F$  présente un saut en  $a$ . Pourquoi ? Comment généreriez-vous  $X$  à partir de la distribution  $F$  tronquée sur  $[a, b]$  dans ce dernier cas ?
2. Expliquez comment générer une variable aléatoire  $X$  pour chacune des distributions suivantes par inversion.
  - (a) La distribution de Pareto de paramètres  $(\alpha, \beta)$ , dont la fonction de distribution est  $F(x) = 1 - (\beta/x)^\alpha$  pour  $x \geq \beta$ .
  - (b) La distribution de Cauchy, pour laquelle  $F(x) = 1/2 + \arctan(x)/\pi$ ,  $x \in \mathcal{R}$ .
  - (c) La distribution logistique, pour laquelle  $F(x) = 1/(1 + e^{-x})$ ,  $x \in \mathcal{R}$ .
  - (d) La distribution de Rayleigh de paramètre  $\beta$ , pour laquelle  $F(x) = 1 - \exp[-x^2/\beta]$ ,  $x > 0$ .
3. Expliquez comment générer
  - une variable aléatoire exponentielle
  - une variable aléatoire géométrique
 en utilisant le principe d'inversion. En utilisant la description d'une variable géométrique en termes d'expériences de Bernoulli, est-il possible de construire un autre générateur pour cette variable discrète ? Si oui, expliquez comment procéder, sinon justifiez votre réponse. Pouvez-vous donner d'avantage de commentaires sur cette dernière approche ?

## Chapitre 6

# Estimation et intervalles de confiance

### 6.1 Estimateurs statistiques

Une simulation ne sert que fort peu si elle n'est pas à même de délivrer des résultats pertinents par rapport aux objectifs de l'étude. Les grandeurs d'intérêt dépendent de la problématique considérée, et des enjeux de l'étude, toutefois, il convient de pouvoir définir comment ces grandeurs peuvent être estimées, et autant que possible, de mesurer leur qualité, d'une manière ou d'une autre. Par exemple, des mesures importantes dans les systèmes de files d'attente peuvent être (Taylor et Karlin [16], Chapitre IX) :

**Nombre de clients dans le système.** Non seulement les clients dans le système impliquent souvent des coûts, mais dans beaucoup de systèmes, de l'espace physique doit être prévu et fourni pour les clients en attente. De plus grands nombres de clients en attente peuvent aussi contrecarrer le processus d'entrée en repoussant de nouveaux clients potentiels.

**Utilisation du ou des serveur(s).** Des serveurs inactifs peuvent induire des coûts sans contribuer à la performance du système.

**Débit de système.** Le nombre de clients passant à long terme à travers le système est une mesure directe de la performance du système.

**Temps d'attente des clients.** De longues attentes avant d'obtenir le service requis sont au mieux ennuyantes dans les systèmes les plus simples, et directement associés avec des coûts majeurs dans nombre de systèmes de plus

grande envergure, comme ceux décrivant par exemple des bateaux attendant de décharger dans un port, ou des patients attendant des soins aux urgences d'un hôpital.

Une fois les quantités à mesurer déterminées, nous allons construire des estimateurs de celles-ci, lesquels estimateurs devraient être aussi précis que possible.

**Definition 6.1:** Estimateur

Un estimateur  $\hat{\mu}$  d'une quantité fixe, mais inconnue,  $\mu$  est un variable ou un vecteur aléatoire qui associe aux données une valeur supposée approcher la véritable valeur  $\mu$ .

## 6.2 Efficacité des estimateurs

Considérons un estimateur  $X$  d'une certaine quantité inconnue  $\mu$ . Le *biais*, la *variance*, l'*erreur quadratique moyenne* (MSE, pour mean square error), et l'*erreur relative* (RE, pour relative error) de  $X$  sont définis respectivement comme suit :

$$\begin{aligned}\beta &= E[X] - \mu; \\ \sigma^2 &= \text{Var}(X) = E[(X - E[X])^2]; \\ \text{MSE}[X] &= E[(X - \mu)^2] = \beta^2 + \sigma^2; \\ \text{RE}[X] &= \sqrt{\text{MSE}[X]} / |\mu|, \text{ pour } \mu \neq 0.\end{aligned}$$

Un estimateur sera dit non-biaisé si  $\beta = 0$ . La racine carrée du  $\text{MSE}[X]$  est appelée l'*erreur absolue* ; c'est une mesure de la précision statistique de l'estimateur  $X$ , et  $\text{RE}[X]$  est une mesure de cette prédiction relativement à l'ordre de grandeur de la moyenne  $\mu$ .

Supposons de plus que l'effort numérique requis pour calculer  $X$  (par exemple en termes de temps CPU) est une variable aléatoire (typiquement corrélée avec  $X$ ) et dénotons son espérance mathématique par  $C(X)$ . L'efficacité de  $X$  est

$$\text{Eff}(X) = \frac{1}{C(X) \cdot \text{MSE}(X)}. \quad (6.1)$$

Un estimateur de  $X$  sera dit être *plus efficace* qu'un autre estimateur  $Y$  si  $\text{Eff}(X) > \text{Eff}(Y)$ .

Dans ce contexte, *améliorer l'efficacité* signifie trouver un estimateur  $Y$  qui est plus efficace que l'estimateur  $X$  actuellement utilisé dans le sens décrit par

(6.1). Souvent, les deux estimateurs sont non biaisés, et sont supposés présenter des temps de calcul similaires. Par conséquent, améliorer l'efficacité revient dans ce cas à réduire la variance. Pour cette raison, nous parlerons souvent de techniques de réduction de variance. Il est toutefois parfois possible d'améliorer l'efficacité en augmentant la variance tout en réduisant le coût de calcul. Si le temps de calcul n'est pas pris en compte, nous appellerons  $\text{Var}[X]/\text{Var}[Y]$  le *facteur de réduction de variance* de  $Y$  par rapport à  $X$ . Il représente le facteur par lequel la variance est réduite en utilisant  $Y$  au lieu de  $X$ .

### 6.3 Intervalles de confiance

Toute mesure de qualité est imparfaite ou incomplète. Ainsi, l'efficacité  $\text{Eff}[X]$  suppose que le coût de l'erreur est symétrique et proportionnel à son carré. Un autre aspect important que  $\text{Eff}[X]$  ne mesure pas est la disponibilité d'une bonne façon d'évaluer l'erreur d'estimation. Par exemple, si on estime cette erreur par la variance de  $X$ , il nous faut un bon estimateur de cette variance. L'évaluation de l'erreur d'estimation est habituellement fournie en donnant un intervalle de confiance (IC), défini comme suit.

**Definition 6.2:** Intervalle de confiance

Un intervalle de confiance  $[I_1, I_2]$  pour une quantité  $\mu$  est un intervalle défini au moyen de deux variables aléatoires  $I_1$  et  $I_2$  satisfaisant  $I_1 \leq I_2$ , donnant une certaine probabilité de contenir  $\mu$ .  $[I_1, I_2]$  est un intervalle de confiance de niveau  $1 - \alpha$  (ou à  $100(1 - \alpha)\%$ ) pour  $\mu$  si  $P[I_1 \leq \mu \leq I_2] = 1 - \alpha$ .

Habituellement, on construit un intervalle de confiance pour un niveau *visé* ou *nominal*  $1 - \alpha$ , mais la véritable probabilité de couverture est différente et inconnue. La différence est l'erreur de couverture. La *largeur* de l'intervalle est  $I_2 - I_1$  (une variable aléatoire). Idéalement, nous voudrions assurer la bonne couverture, tout en conservant  $E[I_2 - I_1]$  et  $\text{Var}[I_2 - I_1]$  petits.

Par abus de notation, nous dénoterons parfois une *suite* d'estimateurs  $\{Y_n, n \geq 1\}$  par  $Y_n$ . Deux exemples classiques sont  $\bar{X}_n$  et  $S_n^2$ . Lorsque  $n \rightarrow \infty$ ,  $Y_n$  est dit *asymptotiquement sans biais* si  $E[Y_n - \mu] \rightarrow 0$ , *consistant* si  $Y_n \rightarrow \mu$  en probabilité, i.e.  $P[|Y_n - \mu| > \epsilon] \rightarrow 0$  pour tout  $\epsilon > 0$ , et *fortement consistant* si  $Y_n \rightarrow \mu$  avec probabilité 1 (ou presque sûrement).  $\bar{X}_n$  et  $S_n^2$  pour sont ainsi fortement consistants par rapport à  $\mu = E[X_i]$ . Un intervalle de confiance  $(I_{n,1}, I_{n,2})$

est asymptotiquement valide si son erreur de couverture converge vers 0.

Les intervalles de confiance considérés ici prennent en compte l'erreur due aux aléas de la simulation, mais pas l'erreur dans l'estimation des paramètres du modèle. Supposons par exemple que dans un certain système, les durées de service sont indépendantes et suivent la loi gamma de paramètres  $(\alpha, \beta)$  inconnus. Supposons de plus que nous disposions de 200 observations de durées de service et que l'on estime  $(\alpha, \beta)$  par  $(\hat{\alpha}, \hat{\beta})$  à partir de ces 200 observations. On suppose pour simplifier que l'on identifie la bonne loi. On utilise ensuite la loi estimée dans un modèle de simulation et on calcule un intervalle de confiance pour une mesure de performance quelconque (par exemple, la durée d'attente moyenne) en faisant  $n$  répétitions de la simulation avec la loi estimée. Si  $n$  tend vers l'infini, la largeur de l'intervalle de confiance tend vers 0, mais l'estimateur converge vers la valeur exacte du modèle avec  $(\hat{\alpha}, \hat{\beta})$ , qui diffère de celle du modèle avec  $(\alpha, \beta)$ . Il y a donc deux sources d'erreur : l'une due au fait que  $n$  est fini, et l'autre due à l'erreur dans les paramètres du modèle. Souvent, il y a plusieurs sources d'erreur de ce second type et elles dominent lorsque  $n$  est grand.

## 6.4 Horizon fini

Supposons que nous observons  $X_1, \dots, X_n$ , des copies i.i.d. de  $X$  obtenues en faisant  $n$  répétitions de la simulation, et que nous voulons estimer  $\mu = E[X]$ . Pour ce faire, nous estimons  $\mu$  par  $\bar{X}_n$  et  $\sigma^2 = \text{Var}[X]$  par  $S_n^2$ .

**Théorème 3.** Si  $X_1, \dots, X_n$  sont i.i.d.  $N(\mu, \sigma^2)$ , alors

- (i)  $\bar{X}_n$  et  $S_n^2$  sont indépendants ;
- (ii)  $(n-1)S_n^2/\sigma^2 \sim \chi^2(n-1)$  ;
- (iii)  $\sqrt{n}(\bar{X}_n - \mu)/S_n \sim \text{Student-}t(n-1)$ .

Ce théorème permet de calculer un intervalle de confiance pour  $\mu$  au niveau  $1 - \alpha$  :

$$(\bar{X}_n \pm t_{n-1, 1-\alpha/2} S_n / \sqrt{n}),$$

où  $P[T_{n-1} \leq t_{n-1, 1-\alpha/2}] = 1 - \alpha/2$ . Lorsque  $n$  est grand, nous pouvons approximer la loi Student- $t$  à  $n-1$  degrés de liberté au moyen d'une  $N(0, 1)$ .

Pour obtenir un intervalle de confiance pour  $\sigma^2$ , on choisira  $x_1$  et  $x_2$  tels que

$$P[x_1 < \chi_{n-1}^2 < x_2] = 1 - \alpha,$$

ce qui permet de poser

$$[I_1, I_2] = [(n-1)S_n^2/x_2, (n-1)S_n^2/x_1].$$

Nous avons alors

$$\begin{aligned} P[I_1 \leq \sigma^2 \leq I_2] &= P[(n-1)S_n^2/x_2 \leq \sigma^2 \leq (n-1)S_n^2/x_1] \\ &= P[x_1 \leq (n-1)S_n^2/\sigma^2 \leq x_2] \\ &= 1 - \alpha. \end{aligned}$$

Ceci n'est valide que si les  $X_i$  suivent la loi normale. La tableau 6.1 explicite les bornes  $(n-1)/x_1$  et  $(n-1)/x_2$  d'un intervalle de confiance sur  $\sigma^2/S_n^2$ . Par exemple, pour  $n = 1000$ , un intervalle de confiance à 90% pour  $\sigma^2$  est donné par

$$[0.930 S_n^2, 1.077 S_n^2]$$

$n$	$\alpha = 0.02$		$\alpha = 0.10$	
	$(n-1)/x_1$	$(n-1)/x_2$	$(n-1)/x_1$	$(n-1)/x_2$
10	0.388	3.518	0.492	2.284
30	0.570	1.939	0.663	1.568
100	0.729	1.413	0.796	1.270
300	0.831	1.216	0.876	1.146
1000	0.902	1.111	0.930	1.077

TABLE 6.1 – Bornes  $(n-1)/x_1$  et  $(n-1)/x_2$  d'un intervalle de confiance sur  $\sigma^2/S_n^2$

Si on a deux échantillons indépendants,  $X_1, \dots, X_m$  i.i.d. normales de variance  $\sigma_x^2$  et  $Y_1, \dots, Y_n$  i.i.d. normales de variance  $\sigma_y^2$ , on peut calculer un intervalle de confiance sur le *rapport des deux variances*, en utilisant le fait que

$$F = \frac{S_{x,m}^2/\sigma_x^2}{S_{y,n}^2/\sigma_y^2} = \frac{S_{x,m}^2 \sigma_y^2}{S_{y,n}^2 \sigma_x^2} \sim F(m-1, n-1),$$

où  $S_{x,m}^2$  et  $S_{y,n}^2$  sont les variances échantillonnales. Si  $P[x_1 < F < x_2] = 1 - \alpha$ , l'intervalle est

$$[I_1, I_2] = \left[ \frac{1}{x_2} \frac{S_{x,m}^2}{S_{y,n}^2}, \frac{1}{x_1} \frac{S_{x,m}^2}{S_{y,n}^2} \right].$$

Ce type d'intervalles est potentiellement utile lorsqu'on estime le *facteur de réduction de variance* entre deux estimateurs.

### 6.4.1 Approximation normale

Lorsque  $n$  est grand,  $\bar{X}_n$  est approximativement normale même si  $X$  ne l'est pas, en vertu du théorème de la limite centrale (TLC). Il existe plusieurs versions du TLC :  $X_i$  de lois différentes, dépendance, TLCs multivariés, TLC fonctionnels, etc. Nous citerons le résultat suivant.

**Théorème 4.** Soient  $X_1, X_2, \dots$  des variables aléatoires indépendantes, avec  $E[X_i] = \mu_i$  et  $\text{Var}[X_i] = \sigma_i^2$ . Posons  $s_n^2 = \sigma_1^2 + \dots + \sigma_n^2$ ,

$$Y_n = \frac{(X_1 - \mu_1) + \dots + (X_n - \mu_n)}{s_n},$$

et  $F_n(x) = P[Y_n \leq x]$ . Alors,  $E[Y_n] = 0$ ,  $\text{Var}[Y_n] = 1$ , et

$$\sup_{n \geq 1, x \in \mathcal{R}} |F_n(x) - \Phi(x)| \leq \kappa \frac{E(|X_1 - \mu_1|^3) + \dots + E(|X_n - \mu_n|^3)}{s_n^3}$$

où  $\kappa = 3$  si les  $X_i$  sont i.i.d. et  $\kappa = 6$  sinon.

La borne sur l'erreur dépend donc de l'asymétrie des distributions. Sous l'hypothèse où  $n$  est suffisamment grand que pour pouvoir approximer la distribution de  $\bar{X}_n$ , nous pourrions choisir pour un intervalle de confiance

$$[\bar{X}_n - z_{1-\alpha/2} S_n / \sqrt{n}, \bar{X}_n + z_{1-\alpha/2} S_n / \sqrt{n}],$$

où  $z_{1-\alpha/2}$  est le quantile  $1 - \alpha/2$  d'une normale  $N(0, 1)$ .

Nous pouvons raisonnablement recourir au TLC pour calculer un intervalle de confiance, sauf si un des cas suivant se présente :

- $n$  est trop petit,
- $\alpha$  est proche de 0,
- les  $X_i$  ont une loi très asymétrique,
- il existe des moments supérieurs très élevés.

**Exemple 26 (Binomiale).** Supposons  $n = 1000$ ,  $X_i \sim \text{Binomiale}(1, p)$ ; on veut estimer  $p$ . Si on a 882 succès,  $\bar{X}_n = 0.882$ . On a alors  $S_n^2 = \bar{X}_n(1 - \bar{X}_n)n/(n - 1) \approx 0.1042$  et un intervalle de confiance à 95% (approximativement) est  $(\bar{X}_n \pm 1.96 S_n / \sqrt{n}) \approx (0.862, 0.902)$ . L'intervalle de confiance ainsi construit nous donne aussi une idée des chiffres significatifs de l'estimateur. Mais si  $\bar{X}_n = 0.998$ , alors on voit que  $p$  est trop proche de 1 et l'approx. normale sera très mauvaise. Dans ce cas, on va plutôt utiliser :  $Y = \sum_{i=1}^n (1 - X_i) \approx \text{Poisson}(n(1 - p))$ .

**Exemple 27.** *Durée de vie d'un système. Soit  $X = \min(G_1, \max(G_2, G_3))$ . Les  $G_j$  sont i.i.d. Weibull ( $\alpha = 0.5, \beta = 1$ ). On simule  $n$  fois, avec  $X_i$  la valeur de  $X$  pour la répétition  $i$ . On calcule un intervalle de confiance à 90% pour  $E[X]$  via le théorème de la limite centrale.*

$n$	Prob. couverture	Estim. $E[I_2 - I_1]/\mu$
5	$0.708 \pm 0.03$	1.16
10	$0.750 \pm 0.03$	0.82
20	$0.800 \pm 0.03$	0.60
40	$0.840 \pm 0.03$	0.44

*Il y a dégradation significative de la couverture. Les  $G_j$  (et les  $X_i$ ) suivent en effet une loi très éloignée de la normale, et on se trompe beaucoup en calculant un intervalle de confiance basé sur la loi normale.*

## 6.5 Intervalle de confiance pour une loi discrète

Soit  $Y$  une variable aléatoire prenant ses valeurs dans  $\{0, 1, 2, \dots\}$  et suivant une loi de paramètre  $\mu$ , telle que  $P_\mu[Y \geq y]$  est croissant en  $\mu$ , où  $P_\mu$  dénote la probabilité quand la valeur du paramètre est  $\mu$ . (Le cas décroissant se traite de manière symétrique.) Des exemples de telles distributions comptent les lois binomiale, géométrique, de Poisson,...

On veut un intervalle de confiance  $[I_1, I_2]$  de niveau (approximatif)  $1 - \alpha$  pour  $\mu$ . Posons  $\alpha = \alpha_1 + \alpha_2$ , avec  $\alpha_1 > 0$  et  $\alpha_2 > 0$ . Nous voudrions  $P[\mu < I_1] \approx \alpha_1$  et  $P[\mu > I_2] \approx \alpha_2$ . Si on observe  $Y = y$ , l'intervalle sera  $[I_1(y), I_2(y)]$ .

*Algorithme :* Prendre pour  $I_1(y)$  et  $I_2(y)$  les solutions de

$$\alpha_1 = P_{I_1}[Y \geq y] \quad \text{et} \quad \alpha_2 = P_{I_2}[Y \leq y]. \quad (6.2)$$

Ceci revient à fixer la probabilité que la variable  $Y$  soit supérieure (respectivement inférieure) à l'observation  $y$ , si le paramètre inconnu était de valeur  $I_1$  (respectivement  $I_2$ ). Dans chacune de ces deux configurations, on s'attend à ce qu'un événement ait une faible probabilité, vu qu'une faible (forte) valeur de  $\mu$  défavorise l'événement considéré, et ce en raison de la monotonie de  $P_\mu[Y \geq y]$  par rapport à  $\mu$ . Nous pouvons résoudre par recherche binaire, par exemple. Pour le cas où  $Y$  est décroissant avec  $\mu$ , il suffit de permuter les signes  $\geq$  et  $\leq$ .

**Proposition 4.** *La probabilité de couverture de cet intervalle est d'au moins  $1 - \alpha$ .*

*Démonstration.* Soit  $y^*(\mu) = \min\{y \in \mathcal{N} : I_1(y) \geq \mu\}$  et  $\nu = I_1(y^*(\mu)) \geq \mu$ . Par définition,  $I_1(y) \geq \mu$  si et seulement si  $y \geq y^*(\mu)$  (autrement dit, le véritable paramètre  $\mu$  serait en dehors, plus précisément à gauche de l'intervalle de confiance construit). Par conséquent, en vertu de la croissance de  $P_\mu$  avec  $\mu$ ,

$$P_\mu[I_1(Y) \geq \mu] \leq P_\nu[I_1(Y) \geq \mu] = P_\nu[Y \geq y^*(\mu)] = \alpha_1.$$

On montre de même que  $P_\mu[I_2(Y) \leq \mu] \leq \alpha_2$ .

Nous avons dès lors

$$\begin{aligned} P_\mu[I_1(Y) \leq \mu \leq I_2(Y)] &= 1 - P[\mu < I_1(Y) \cup I_2(Y) < \mu] \\ &= 1 - P_\mu[\mu < I_1(Y)] - P_\mu[I_2(Y) < \mu] \\ &\geq 1 - \alpha_1 - \alpha_2 = 1 - \alpha. \end{aligned}$$

□

La probabilité de couverture *exacte* dépend de  $F_\mu$  et est généralement inconnue.

**Exemple 28.** Reprenons l'exemple 26. En particulier, supposons que  $X_1, \dots, X_n$  sont i.i.d. avec  $P[X_i = 1] = 1 - P[X_i = 0] = p$ , de sorte que  $Y = n\bar{X}_n = \sum_{i=1}^n X_i$  suit une binomiale  $(n, p)$ . Nous souhaitons calculer un intervalle de confiance sur  $p$  basé sur l'observation de  $Y$ . Pour n'importe quelles valeurs de  $p$  et de  $y$ , les probabilités dans (6.2) peuvent être calculées en sommant les probabilités binomiales exactes si  $y$  est petit. Si  $n$  est grand et  $p$  est petits,  $Y$  est approximativement une variable aléatoire de Poisson de moyenne  $np$ , aussi peut-on approximer les probabilités dans (6.2) en additionnant les probabilités de Poisson appropriées. Pour  $p$  proche de 1, nous pouvons with mean  $np$ , so one can approximate the probabilities in (5.1) by summing the appropriate Poisson probabilities. Pour  $p$  proche de 1, nous pouvons simplement remplacer  $p$  et  $X_i$  par  $1 - p$  et  $1 - X_i$ .

## 6.6 Estimation séquentielle

Pour un intervalle de confiance de niveau  $1 - \alpha$ , si on fixe  $n$ , la largeur  $I_2 - I_1$  est aléatoire. Si on veut  $I_2 - I_1 \leq w$  pour  $w$  fixé, la valeur minimale de  $n$  requise est une variable aléatoire  $N$ . Comment prédire ce  $N$  ?

**Exemple 29.** Pour  $X_i \sim \text{binomiale}(1, p)$ , avec  $n = 1000$  on a obtenu  $\bar{X}_n = 0.882$ ,  $S_n^2 \approx 0.1042$ , et la demi-largeur du intervalle de confiance à 95% était de

0.020. Combien de répétitions additionnelles faut-il pour réduire la demi-largeur à environ 0.005 ?

Nous voulons  $1.96S_n/\sqrt{n} \leq 0.005$ . En supposant que  $S_n$  ne changera pas significativement, cela donne  $n \geq (1.96 \times S_n/0.005)^2 \approx 16011.8$ . En conséquence, nous pouvons recommander de faire 15012 répétitions additionnelles.

### 6.6.1 Procédure à deux étapes

Cette approche est valable pour la loi de Student (ou normale). Faire  $n_0$  répétitions et calculer  $S_{n_0}^2$  ; la prédiction du  $n$  requis est

$$\hat{N}^* = \min \{n \mid (t_{n-1, 1-\alpha/2})S_{n_0}/\sqrt{n} \leq r\}.$$

On fera  $\max(0, \hat{N}^* - n_0)$  répétitions additionnelles.

Bien sûr, il se peut que ce soit insuffisant, ou trop.

### 6.6.2 Estimation séquentielle

Après  $n_0$ , recalculer  $S_n^2$  et la demi-largeur pour chaque  $n$ . On s'arrête dès que  $I_2 - I_1 \leq w$ .

Cette procédure est biaisée, car on tend à s'arrêter à un  $N$  où  $S_N^2$  sous-estime la variance. Mais lorsque  $w \rightarrow 0$ , le biais disparaît,  $N/n^* \rightarrow 1$  a.p.1 où  $n^*$  est la valeur optimale de  $N$  si on connaissait  $\sigma^2$ , et  $P[|\bar{X}_N - \mu| \leq w/2] \rightarrow 1 - \alpha$ .

## 6.7 Estimation de quantiles

Si  $X$  est de répartition  $F$ , le  $q$ -quantile de  $F$  est

$$\xi_q = F^{-1}(q) = \inf\{x : F(x) \geq q\}.$$

Soit  $X_{(1)}, \dots, X_{(n)}$  un échantillon i.i.d. de  $X$ , trié, et  $\hat{F}_n$  la fonction de répartition empirique. Un estimateur simple de  $\xi_q$  est le *quantile empirique*

$$\hat{\xi}_{q,n} = \hat{F}_n^{-1}(q) = \inf\{x : \hat{F}_n(x) \geq q\} = X_{(\lceil nq \rceil)}.$$

Il est biaisé mais fortement consistant et obéit au théorème de la limite centrale, comme le montre le théorème ci-dessous.

**Theorème 5.** (i) Pour chaque  $q$ ,  $\hat{\xi}_{q,n} \xrightarrow{\text{p.s.}} \xi_q$  quand  $n \rightarrow \infty$ .

(ii) Si  $X$  a une densité  $f$  strictement positive et continue dans un voisinage de  $\xi_q$ , alors

$$\frac{\sqrt{n}(\hat{\xi}_{q,n} - \xi_q)f(\xi_q)}{\sqrt{q(1-q)}} \Rightarrow N(0, 1) \quad \text{quand } n \rightarrow \infty.$$

Ce TLC indique qu'il y a beaucoup de bruit (variance) si  $f(\xi_q)$  est petit. De plus, pour l'utiliser afin de construire un intervalle de confiance, il faut estimer  $f(\xi_q)$ , ce qui est difficile. Nous pouvons néanmoins construire une méthode non-asymptotique de calcul d'un intervalle de confiance pour  $\xi_q$  : supposons que  $F$  est continue en  $\xi_q$ . Soit  $B$  le nombre d'observations  $X_{(i)}$  inférieures à  $\xi_q$ . Puisque  $P[X < \xi_q] = q$ ,  $B$  est binomiale( $n, q$ ). Si  $1 \leq j < k \leq n$ ,  $X_{(j)} < \xi_q \leq X_{(k)}$  ssi  $j \leq B < k$ . Alors

$$P[X_{(j)} < \xi_q \leq X_{(k)}] = P[j \leq B < k] = \sum_{i=j}^{k-1} \binom{n}{i} q^i (1-q)^{n-i}.$$

On choisit  $j$  et  $k$  pour que cette somme soit supérieure ou égale à  $1 - \alpha$  (intervalle unilatéral ou bilatéral).

Si  $n$  est grand et  $q$  n'est pas trop proche de 0 ou 1, on peut approximer la loi binomiale par la loi normale :

$$\frac{B - nq}{\sqrt{nq(1-q)}} \approx N(0, 1).$$

On obtient alors  $j = \lfloor nq + 1 - \delta \rfloor$  et  $k = \lfloor nq + 1 + \delta \rfloor$ , où  $\delta = \sqrt{nq(1-q)}\Phi^{-1}(1 - \alpha/2)$ .

**Exemple 30** (Valeur à risque). Soit  $L$  la perte nette de valeur d'un porte-feuille d'actifs pour une période de temps donnée  $[0, T]$ . La valeur à risque (VAR) (au temps 0) est la valeur de  $x_p$  telle que  $P[L > x_p] = p$ . C'est le  $(1 - p)$ -quantile de  $L$ .

Valeurs courantes :  $p = 0.01$ ,  $T = 2$  semaines (banques),  $T =$  mois ou années (assurance, fonds de pension). On peut critiquer l'utilisation de la VAR, vu qu'elle donne une information très limitée. Par exemple si  $x_{0.01} = 10^7$  dollars, que sait-on sur l'importance réelle de la perte ? Une mesure complémentaire pourrait être  $E[L \mid L > x_p]$ , par exemple.

Modèles pour estimer la VAR : on doit modéliser l'évolution du prix des actifs (souvent plusieurs milliers, dépendants). Souvent : modèles à facteurs. On peut

## 6.8. INTERVALLE DE CONFIANCE PAR RÉÉCHANTILLONNAGE (“BOOTSTRAP”)93

*remplacer les actifs par des prêts, comptes à payer, etc. Sauf dans les cas simples, on estime la VAR par simulation. Quand  $p$  est petit : “importance sampling”.*

### 6.8 intervalle de confiance par rééchantillonnage (“bootstrap”)

Il s’agit de techniques de simulation appliquées en statistique ; l’idée est d’estimer la distribution (inconnue et quelconque) de l’estimateur en rééchantillonnant des échantillons de taille  $n$  en tirant avec remplacement dans l’échantillon de taille  $n$  original. Pour chaque échantillon ainsi construit, nous recalculons l’estimateur en cours d’étude. Pour plus de détails, le lecteur pourra consulter par exemple Efron et Tibshirani [6], dont s’inspire largement cette section.

#### 6.8.1 Principe de plug-in

Considérons un échantillon i.i.d.  $\mathbf{X} = X_1, \dots, X_n$ , issu d’une loi de fonction de répartition  $F$ , et un estimateur  $Y = g(X_1, \dots, X_n)$  d’une valeur réelle inconnue  $\theta$ . Par exemple,  $Y = \bar{X}_n$  avec  $\theta = \mu$ , ou  $Y = S_n^2$  avec  $\theta = \sigma^2$ , etc.  $Y$  peut être biaisé (i.e.  $E[Y] \neq \theta$ ), mais nous supposons que  $g$  ne dépend pas de l’ordre des  $X_i$ ’s. Si nous ne connaissons pas la distribution exacte  $F$ , nous pouvons toujours nous diriger vers la distribution empirique construite à partir de l’échantillon  $\mathbf{X}$ . Le principe de plug-in est une méthode simple d’estimation de paramètre à partir d’échantillons. L’estimateur plug-in d’un paramètre  $\theta = g(F)$  est défini comme

$$\hat{\theta} = g(\hat{F}).$$

En général, l’estimateur plug-in d’une espérance  $\theta = E_F(x)$  est

$$E_{\hat{F}} = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x},$$

autrement dit nous retrouvons l’estimateur de moyenne classique.

A quel point le principe de plug-in est-il pertinent ? Il est en général assez bon, si la seule source d’information disponible à propos de  $F$  vient de l’échantillon  $\mathbf{X}$ . Sous cette circonstance,  $\hat{\theta}_n = g(\hat{F}_n)$  ne peut pas être amélioré comme estimateur de  $\theta = g(F)$ , du moins pas dans le sens asymptotique habituel en théorie

statistique. Par exemple, si  $\hat{f}_k$  est l'estimateur de fréquence plug-in  $\#\{x_i = k\}/n$ , alors

$$\hat{f}_k \sim \text{Bi}(n, f_k)/n.$$

Dans ce cas, l'estimateur  $\hat{f}_k$  est non-biaisé pour  $f_k$ ,  $E[\hat{f}_k] = f_k$ , de variance  $f_k(1 - f_k)/n$ . Il s'agit de la plus petite variance possible pour un estimateur sans biais de  $f_k$ .

Nous allons voir que le bootstrap lui-même est une application du principe de plug-in, en supposons que nous ne disposons que d'un seul échantillon aléatoire  $X$  d'une distribution totalement inconnue  $F$ .

### 6.8.2 Bootstrap non-paramétrique

Considérons à présent  $K_n(F, z) = P[Y - \theta \leq z]$  pour  $z \in \mathcal{R}$ . Un intervalle de confiance *exact* pour  $\theta$ , au niveau  $1 - \alpha_1 - \alpha_2$ , est

$$(I_1, I_2) = (Y - K_n^{-1}(F, 1 - \alpha_1), Y - K_n^{-1}(F, \alpha_2)),$$

où  $K_n^{-1}(F, q)$  est le  $q$ -quantile de  $K_n(F, \cdot)$ . En effet,  $P[I_1 > \theta] = P[Y - \theta > K_n^{-1}(F, 1 - \alpha_1)] = 1 - K_n(F, K_n^{-1}(F, 1 - \alpha_1)) = \alpha_1$ . De même, nous avons  $P[I_2 < \theta] = \alpha_2$ .

Toutefois, il est rare de connaître  $K_n(F, \cdot)$ . Une première idée serait de répéter l'expérience  $m$  fois afin obtenir  $m$  copies i.i.d. de  $Y$  pour estimer sa distribution. Si  $E[Y] = \theta$ , on peut estimer ainsi la distribution de  $Y - \theta$ . Mais cela ferait  $mn$  simulations ! Souvent, il est très coûteux, voire même impossible, d'avoir de nouvelles copies de  $Y$ . L'idée du bootstrap consiste à remplacer  $F$  par  $\hat{F}_n$  et  $\theta$  par  $y$  dans  $K_n(F, z)$ . Soient  $x_1, \dots, x_n$  les valeurs de  $X_1, \dots, X_n$  et  $y = g(x_1, \dots, x_n)$ .

### 6.8.3 Bootstrap non-paramétrique de base

Tirons  $X_1^*, \dots, X_n^*$  au hasard avec remplacement de l'échantillon de départ  $\{x_1, \dots, x_n\}$  (i.e., de  $\hat{F}_n$ ) et calculons  $Y^* = g(X_1^*, \dots, X_n^*)$ . L'opération est répétée  $m$  fois, de sorte que nous obtenions  $m$  copies i.i.d. de  $Y^*$ , à savoir  $Y_1^*, \dots, Y_m^*$ . Comme indiqué précédemment, cela revient à répéter l'expérience  $m$  fois avec  $\hat{F}_n$  au lieu de  $F$ . La notation étoile indique que  $x^*$  n'est pas l'ensemble de données réel  $x$ , mais plutôt une version randomisée, ou rééchantillonnée, de  $x$ . L'algorithme 6.1 illustre l'idée du bootstrap pour l'estimation d'écart-type.

## 6.8. INTERVALLE DE CONFIANCE PAR RÉÉCHANTILLONNAGE (“BOOTSTRAP”)95

**Algorithm 6.1:** Algorithme bootstrap d’estimation d’écart-type

1. Tirer  $m$  échantillons bootstrap indépendants  $\mathbf{x}_1^*, \mathbf{x}_2, \dots, \mathbf{x}_m^*$ , chacun consistant de  $n$  valeurs de données tirées avec remplacement de  $\mathbf{x}$ .
2. Evaluer la réplique bootstrap correspondante à chaque échantillon bootstrap,

$$\hat{\theta}^*(i) = g(x_i^*), \quad i = 1, 2, \dots, m.$$

3. Estimer l’erreur standard  $se_F(\hat{\theta})$  par l’écart-type échantillonnal des  $m$  répliques :

$$\hat{se}_m = \sqrt{\frac{1}{m-1} \sum_{i=1}^m (\hat{\theta}^*(i) - \hat{\theta}^*(\cdot))^2},$$

$$\text{où } \hat{\theta}^*(\cdot) = \frac{1}{m} \sum_{i=1}^m \hat{\theta}^*(i).$$

La limite de  $\hat{se}_m$  comme  $m$  tend vers l’infini est l’estimateur bootstrap idéal de  $se_F(\hat{\theta})$ ,

$$\lim_{m \rightarrow \infty} \hat{se}_m = se_{\hat{F}} = se_{\hat{F}}(\hat{\theta}^*).$$

L’estimateur de bootstrap idéal  $se_{\hat{F}}(\hat{\theta}^*)$  et son approximation  $\hat{se}_m$  sont parfois appelés estimateurs bootstrap non-paramétriques car ils sont basés sur  $\hat{F}$ , l’estimateur non-paramétrique de la population  $F$ .

Soit  $\hat{K}_{n,m}$  la fonction de répartition empirique de  $Y_1^* - y, \dots, Y_m^* - y$ . Pour  $m \rightarrow \infty$ , elle converge vers la fonction de répartition de  $Y^* - y$ , qui est  $K_n(\hat{F}_n, \cdot)$ . L’intervalle de confiance retourné est :

$$(y - \hat{K}_{n,m}^{-1}(1 - \alpha_1), y - \hat{K}_{n,m}^{-1}(\alpha_2)) = (2y - Y_{(\lceil m(1-\alpha_1) \rceil)}^*, 2y - Y_{(\lceil m\alpha_2 \rceil)}^*).$$

Cela revient à remplacer  $F$  par  $\hat{F}_n$  puis à approximer  $K_n(\hat{F}_n, \cdot)$  par  $\hat{K}_{n,m}$ . Il y a donc deux sources d’erreur, qui sont cependant la plupart du temps inévitables.

### 6.8.4 Bootstrap-t non-paramétrique

Supposons que nous disposons également d’un estimateur de la variance de  $Y$ , disons  $S^2 = h^2(X_1, \dots, X_n)$ . Soit  $J_n(F, \cdot)$  la fonction de répartition de la statistique *studentisée*  $(Y - \theta)/S$ . Un intervalle de confiance exact de niveau  $(1 - \alpha_1 - \alpha_2)$  :

$$(I_1, I_2) = (Y - J_n^{-1}(F, 1 - \alpha_1)S, Y - J_n^{-1}(F, \alpha_2)S).$$

L'algorithme du bootstrap- $t$  non-paramétrique consiste, pour chacune des  $m$  répétitions bootstrap, à générer  $n$  observations  $X_1^*, \dots, X_n^*$  comme avant, puis à calculer  $Y^* = g(X_1^*, \dots, X_n^*)$ ,  $S^* = h(X_1^*, \dots, X_n^*)$ , et  $Z^* = (Y^* - y)/S^*$ .

Soient  $Z_1^*, \dots, Z_m^*$  les  $m$  copies i.i.d. de  $Z^*$  et  $\hat{J}_{n,m}$  leur fonction de répartition empirique. Pour calculer l'intervalle de confiance, on remplace  $J_n(F, \cdot)$  par  $\hat{J}_{n,m}(\cdot)$  :

$$\begin{aligned} (I_1, I_2) &= (y - \hat{J}_{n,m}^{-1}(1 - \alpha_1)S, y - \hat{J}_{n,m}^{-1}(\alpha_2)S) \\ &= (y - Z_{(\lceil m(1-\alpha_1) \rceil)}^* S, y - Z_{(\lceil m\alpha_2 \rceil)}^* S). \end{aligned}$$

Empiriquement, le bootstrap- $t$  performe souvent le mieux. Le choix de  $m$  influence peu l'erreur de couverture, mais un trop petit  $m$  donne des intervalles de confiance dont la largeur varie beaucoup. Un choix populaire consiste à prendre  $m = 1000$ .

### 6.8.5 Estimation du biais

Une application particulièrement intéressante du bootstrap est la possibilité d'estimer le biais d'un estimateur quelconque. Sous la distribution  $F$ , le biais d'un estimateur  $\hat{\theta} = g(\mathbf{X})$  d'une quantité inconnue  $\theta = t(F)$  est défini comme

$$B_F(\hat{\theta}, \theta) = E_F[g(\mathbf{X})] - t(F).$$

L'estimateur bootstrap de biais est défini comme

$$B_{\hat{F}}(\hat{\theta}, \theta) = E_{\hat{F}}[g(\mathbf{X}^*)] - t(\hat{F}).$$

L'estimateur plug-in  $t(\hat{F})$  de  $\theta$  peut différer de  $\hat{\theta} = g(x)$ . En d'autres termes,  $B_{\hat{F}}(\hat{\theta}, \theta)$  est l'estimateur plug-in de  $B_F(\hat{\theta}, \theta)$ , que  $\hat{\theta}$  soit ou non l'estimateur plug-in de  $\theta$ . Dans la plupart des cas,  $E_{\hat{F}}[g(\mathbf{X}^*)]$  devra être approximé par simulation Monte-Carlo :

$$\hat{\theta}^* = \frac{1}{m} \sum_{i=1}^m \theta^*(i) = \frac{1}{m} \sum_{i=1}^m g(\mathbf{x}_i^*).$$

L'estimateur de bootstrap de biais basé sur les  $m$  répliques bootstrap est

$$\hat{B}_m = \hat{\theta}^* - t(\hat{F}).$$

Il est possible d'améliorer cet estimateur quand  $\hat{\theta}$  est l'estimateur plug-in  $t(\hat{F})$  de  $\theta = t(F)$ . Soit  $P_j^*$  la proportion du  $j^e$  point de données originales dans l'échantillon bootstrap  $\mathbf{x}^* = \{x_1^*, x_2^*, \dots, x_n^*\}$  :

$$P_j^* = \frac{\#\{x_i^* = x_j\}}{n}, \quad j = 1, 2, \dots, n.$$

Le vecteur de rééchantillonnage

$$\mathbf{P}^* = (P_1^*, P_2^*, \dots, P_n^*)$$

a des composantes non-négatives dont la somme est égale à 1. Une réplication bootstrap  $\hat{\theta}^*$  peut être vue comme une fonction du vecteur de rééchantillonnage  $\mathbf{P}^*$ . Pour  $\hat{\theta} = t(\hat{F})$ , l'estimateur plug-in de  $\theta$ , nous écrivons

$$\hat{\theta}^* = T(\mathbf{P}^*)$$

pour indiquer que  $\hat{\theta}^*$  est une fonction du vecteur de rééchantillonnage. Les  $m$  échantillons bootstrap  $\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_m^*$  donnent lieu aux vecteurs de rééchantillonnage correspondants  $P_1^*, P_2^*, \dots, P_m^*$ . Définissons  $\bar{\mathbf{P}}^*$  comme la moyenne de ces vecteurs :

$$\bar{\mathbf{P}}^* = \frac{1}{m} \sum_{i=1}^m \mathbf{P}_i^*.$$

En écrivant

$$\mathbf{P}_0 = \left( \frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n} \right),$$

l'estimateur de biais bootstrap devient

$$\hat{B}_m = \hat{\theta}^* - T(\mathbf{P}_0).$$

L'estimateur de bootstrap amélioré est défini comme

$$\bar{B}_m = \hat{\theta}^* - T(\bar{\mathbf{P}}^*).$$

$\hat{B}_m$  et  $\bar{B}_m$  convergent vers  $B_{\hat{F}}$ , toutefois il est possible de montrer que la convergence est plus rapide pour  $\bar{B}_m$ .

Il est toutefois dangereux d'utiliser ces estimations de biais pour corriger l'estimateur  $\hat{\theta}$ , car ils ajoutent de la variance à ce dernier.

## 6.9 Intervalle de confiance pour une fonction de plusieurs moyennes

Dans le cas déterministe, nous savons que si  $\mathbf{Y}_n = (Y_{1n}, \dots, Y_{dn})$  converge vers un certain vecteur  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_d)$  et si  $g : \mathcal{R}^d \rightarrow \mathcal{R}$  est continue, alors  $g(\mathbf{Y}_n) \rightarrow g(\boldsymbol{\mu})$ . Supposons à présent que les  $\mathbf{Y}_n$  sont des vecteurs aléatoires et que  $r(n)(\mathbf{Y}_n - \boldsymbol{\mu}) \Rightarrow \mathbf{Y}$ . Par exemple, si  $\mathbf{Y}_n$  est une moyenne de  $n$  vecteurs,

nous savons que  $\sqrt{n}(\mathbf{Y}_n - \boldsymbol{\mu}) \Rightarrow N(\mathbf{0}, \boldsymbol{\Sigma}_y)$ . Avons-nous encore la convergence de  $r(n)(g(\mathbf{Y}_n) - g(\boldsymbol{\mu}))$ , et le cas échéant, vers quelle distribution ? La réponse est fournie par le théorème Delta.

**Théorème 6** (Théorème Delta). *Soit  $g : \mathcal{R}^d \rightarrow \mathcal{R}$  continûment différentiable dans un voisinage de  $\boldsymbol{\mu}$ , et  $\nabla g$  son gradient. Si  $r(n)(\mathbf{Y}_n - \boldsymbol{\mu}) \Rightarrow \mathbf{Y}$  quand  $n \rightarrow \infty$ , alors*

$$r(n)(g(\mathbf{Y}_n) - g(\boldsymbol{\mu})) \Rightarrow (\nabla g(\boldsymbol{\mu}))^T \mathbf{Y} \quad \text{quand } n \rightarrow \infty.$$

**Corollaire 1** (Corollaire). *Si  $\sqrt{n}(\mathbf{Y}_n - \boldsymbol{\mu}) \Rightarrow N(\mathbf{0}, \boldsymbol{\Sigma}_y)$  quand  $n \rightarrow \infty$ , alors on a le TLC :*

$$\sqrt{n}(g(\mathbf{Y}_n) - g(\boldsymbol{\mu}))/\sigma_g \Rightarrow N(0, 1) \quad \text{quand } n \rightarrow \infty,$$

où  $\sigma_g^2 = (\nabla g(\boldsymbol{\mu}))^T \boldsymbol{\Sigma}_y \nabla g(\boldsymbol{\mu})$ .

**Exemple 31** (Quotient de deux espérances). *Soient  $(X_1, Y_1), \dots, (X_n, Y_n)$  des copies i.i.d. de  $(X, Y)$  et supposons que l'on estime  $\nu = E[X]/E[Y]$  par*

$$\hat{\nu}_n = \frac{\bar{X}_n}{\bar{Y}_n} = \frac{\sum_{i=1}^n X_i}{\sum_{i=1}^n Y_i}.$$

*Cet estimateur est biaisé mais fortement consistant.*

*Posons  $\mu_1 = E[X]$ ,  $\mu_2 = E[Y]$ ,  $g(\mu_1, \mu_2) = \mu_1/\mu_2$ ,  $\sigma_1^2 = \text{Var}[X]$ ,  $\sigma_2^2 = \text{Var}[Y]$ , et  $\sigma_{12} = \text{Cov}[X, Y]$ . Supposons que ces quantités soient finies et que  $\mu_2 \neq 0$ ,  $\sigma_1^2 > 0$ , et  $\sigma_2^2 > 0$ . Le gradient de  $g$  est*

$$\nabla g(\mu_1, \mu_2) = (1/\mu_2, -\mu_1/\mu_2^2)^T.$$

*En vertu du théorème de la limite centrale, nous pouvons en outre écrire*

$$\sqrt{n}(\bar{X}_n - \mu_1, \bar{Y}_n - \mu_2)^T \Rightarrow (W_1, W_2)^T \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

où

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}.$$

*Puis, par le théorème delta (ou son corollaire), nous avons*

$$\sqrt{n}(\hat{\nu}_n - \nu) \Rightarrow (W_1, W_2) \cdot \nabla g(\mu_1, \mu_2) = W_1/\mu_2 - W_2\mu_1/\mu_2^2 \sim N(0, \sigma_g^2)$$

où

$$\begin{aligned} \sigma_g^2 &= (\nabla g(\boldsymbol{\mu}))^T \boldsymbol{\Sigma}_y \nabla g(\boldsymbol{\mu}) \\ &= \sigma_1^2/\mu_2^2 + \sigma_2^2\mu_1^2/\mu_2^4 - 2\sigma_{12}\mu_1/\mu_2^3, \end{aligned}$$

## 6.9. INTERVALLE DE CONFIANCE POUR UNE FONCTION DE PLUSIEURS MOYENNES 99

ou encore

$$\sigma_g^2 = \frac{\sigma_1^2 + \sigma_2^2 \nu^2 - 2\sigma_{12}\nu}{\mu_2^2}. \quad (6.3)$$

Nous pouvons calculer un intervalle de confiance en utilisant ce dernier théorème de la limite centrale si nous disposons d'un bon estimateur de  $\sigma_g^2$ . Un candidat évident est :

$$\hat{\sigma}_{g,n}^2 = \frac{\hat{\sigma}_1^2 + \hat{\sigma}_2^2 \hat{\nu}_n^2 - 2\hat{\sigma}_{12}\hat{\nu}_n}{\bar{Y}_n^2},$$

où

$$\begin{aligned} \hat{\sigma}_1^2 &= \frac{1}{n-1} \sum_{j=1}^n (X_j - \bar{X}_n)^2, \\ \hat{\sigma}_2^2 &= \frac{1}{n-1} \sum_{j=1}^n (Y_j - \bar{Y}_n)^2, \\ \hat{\sigma}_{12} &= \frac{1}{n-1} \sum_{j=1}^n (X_j - \bar{X}_n)(Y_j - \bar{Y}_n). \end{aligned}$$

Puisque  $\hat{\sigma}_{g,n}^2$  est fortement consistant, on obtient le théorème de la limite centrale

$$\frac{\sqrt{n}(\hat{\nu}_n - \nu)}{\hat{\sigma}_{g,n}} \Rightarrow \frac{\sqrt{n}(\hat{\nu}_n - \nu)}{\sigma_g} \Rightarrow N(0, 1) \quad \text{quand } n \rightarrow \infty.$$

L'intervalle de confiance classique pour  $\nu$  au niveau nominal  $1-\alpha$  est  $(\hat{\nu}_n - r, \hat{\nu}_n + r)$  où  $r = z_{1-\alpha/2} \hat{\sigma}_{g,n} / \sqrt{n}$ .

Son erreur de couverture est parfois grande lorsque  $n$  n'est pas très grand, ou lorsque la convergence vers  $N(0, 1)$  est lente. Dans ce cas, on recommande d'utiliser le bootstrap- $t$  non-paramétrique, en prenant  $\hat{\nu}_n$  et  $\hat{\sigma}_{g,n}$  comme estimateurs de la moyenne et de la variance.

Pour le cas particulier d'un rapport de deux espérances, la dérivation suivante est plus directe. Les variables aléatoires

$$Z_j = X_j - \nu Y_j,$$

sont i.i.d. de moyenne 0 et de variance

$$\begin{aligned} \sigma_z^2 &= \text{Var}[Z_j] = \text{Var}[X_j] + \nu^2 \text{Var}[Y_j] - 2\nu \text{Cov}(X_j, Y_j) \\ &= \sigma_1^2 + \sigma_2^2 \nu^2 - 2\sigma_{12}\nu. \end{aligned}$$

En appliquant le TLC aux  $Z_j$ , on obtient

$$\frac{\sqrt{n}\bar{Y}_n(\hat{\nu}_n - \nu)}{\sigma_z} = \frac{\sqrt{n}\bar{Z}_n}{\sigma_z} \Rightarrow N(0, 1) \quad \text{quand } n \rightarrow \infty.$$

C'est équivalent, car  $\sigma_z/\bar{Y}_n \xrightarrow{\text{p.s.}} \sigma_z/\mu_2 = \sigma_g$  quand  $n \rightarrow \infty$ .

Remarque importante : on préfère  $\text{Cov}(X_j, Y_j) > 0$  !

### 6.9.1 Différence entre deux moyennes

On a  $n_1$  observations i.i.d.  $X_{11}, \dots, X_{1,n_1}$ , de moyenne  $\mu_1$ , et  $n_2$  observations i.i.d.  $X_{21}, \dots, X_{2,n_2}$ , de moyenne  $\mu_2$ . On veut un intervalle de confiance pour  $\mu_1 - \mu_2$ .

Les deux méthodes suivantes supposent que les  $X_{ji}$  suivent la *loi normale*. (Pas toujours valide !) Dans la seconde (Welch), les deux échantillons doivent être indépendants mais on peut avoir  $n_1 \neq n_2$ . Dans la première, il faut  $n_1 = n_2$  mais  $X_{1i}$  et  $X_{2i}$  peuvent être corrélés.

#### 1. Observations couplées.

Soit  $n_1 = n_2 = n$ . Posons  $Z_i = X_{1i} - X_{2i}$  pour  $1 \leq i \leq n$ ,

$$\bar{Z}_n = \frac{1}{n} \sum_{i=1}^n Z_i, \quad \text{et} \quad S_n^2 = \frac{1}{n-1} \sum_{i=1}^n (Z_i - \bar{Z})^2.$$

Puisque les  $Z_i$  sont i.i.d. normales,

$$\sqrt{n}[\bar{Z}_n - (\mu_1 - \mu_2)]/S_n \sim \text{Student}(n-1).$$

On utilise cela pour calculer l'intervalle de confiance. Puisque

$$\text{Var}[Z_i] = \text{Var}[X_{1i}] + \text{Var}[X_{2i}] - 2\text{Cov}[X_{1i}, X_{2i}],$$

il est avantageux d'avoir  $\text{Cov}[X_{1i}, X_{2i}] > 0$ .

#### 2. Méthode de Welch.

On suppose que  $X_{1i}$  et  $X_{2i}$  sont indépendants. Soit

$$\bar{X}_{(k)} = \frac{1}{n_k} \sum_{i=1}^{n_k} X_{ki} \quad \text{et} \quad S_{(k)}^2 = \frac{1}{n_k-1} \sum_{i=1}^{n_k} (X_{ki} - \bar{X}_{(k)})^2,$$

pour  $k = 1, 2$ . Alors,

$$\frac{\bar{X}_{(1)} - \bar{X}_{(2)} - (\mu_1 - \mu_2)}{[S_{(1)}^2/n_1 + S_{(2)}^2/n_2]^{1/2}} \approx \text{Student}(\hat{\ell})$$

où

$$\hat{\ell} = \frac{[S_{(1)}^2/n_1 + S_{(2)}^2/n_2]^2}{[S_{(1)}^2/n_1]^2/(n_1 - 1) + [S_{(2)}^2/n_2]^2/(n_2 - 1)}.$$

## 6.10 Moyenne sur horizon infini, état stationnaire

Soit  $\{C_i, i \geq 1\}$  un processus faiblement stationnaire (A.2, page 159), caractérisé par les propriétés suivantes :  $E[C_i] = \mu$ ,  $\text{Var}[C_i] = \sigma^2 > 0$ , coefficients d'autocorrelations  $\rho_k = \rho_{i,i+k}$ . Nous souhaitons estimer  $\mu$  par la moyenne empirique

$$\bar{C}_n = \frac{1}{n} \sum_{i=1}^n C_i.$$

On a  $E[\bar{C}_n] = \mu$  et

$$\text{Var}[\bar{C}_n] = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \text{Cov}[C_i, C_j] = \frac{\sigma^2(1 + \gamma_n)}{n} = \frac{\sigma_n^2}{n}$$

où

$$\gamma_n = \frac{2}{n} \sum_{i=1}^n \sum_{j=i+1}^n \rho_{j-i} = \frac{2}{n} \sum_{k=1}^{n-1} (n-k) \rho_k.$$

Soit

$$\gamma = \lim_{n \rightarrow \infty} \gamma_n = 2 \sum_{k=1}^{\infty} \rho_k$$

si la limite existe. Si  $\gamma < \infty$ , alors

$$\text{Var}[\bar{C}_n] = \frac{\sigma^2(1 + \gamma_n)}{n} = \frac{\sigma_\infty^2}{n} + o(1/n)$$

où

$$\sigma_\infty^2 = \lim_{n \rightarrow \infty} \sigma_n^2 = \lim_{n \rightarrow \infty} n \text{Var}[\bar{C}_n] = \sigma^2(1 + \gamma).$$

La variance est gonflée par un facteur  $\sigma_\infty^2/\sigma^2 = 1 + \gamma$  par rapport au cas des  $C_i$  indépendants. On peut voir  $\gamma$  comme une mesure globale d'autocorrélation.

Si  $\hat{\sigma}_n^2$  est un estimateur consistant pour  $\sigma_\infty^2$ , on s'attend à avoir le TLC :

$$\frac{\sqrt{n}(\bar{C}_n - \mu)}{\hat{\sigma}_n} \Rightarrow \frac{\sqrt{n}(\bar{C}_n - \mu)}{\sigma_\infty} \Rightarrow N(0, 1) \text{ quand } n \rightarrow \infty.$$

(a) Sous quelles conditions a-t-on ce TLC ?

(b) Comment construire  $\hat{\sigma}_n^2$  ?

**Exemple 32** (File  $M/M/1$ ). Taux d'arrivée  $\lambda$ , taux de service  $\mu$ .

La loi d'équilibre des temps d'attentes  $\{W_j, j \geq 1\}$  est donnée par  $F_w(x) = P[W_j \leq x] = 1 - \rho e^{-(\mu-\lambda)x}$  pour  $x \geq 0$ , où  $\rho = \lambda/\mu$ . Si  $W_1$  suit cette loi, chaque  $W_j$  aussi et on a

$$w = E[W_j] = \frac{\rho}{\mu(1-\rho)} \quad \text{et} \quad \sigma^2 = \text{Var}[W_j] = \frac{\rho(2-\rho)}{\mu^2(1-\rho)^2}.$$

Il est important de noter que ce n'est pas équivalent à démarrer le processus à  $W_1 = w$ .

On a aussi une formule pour  $\rho_j$ ,  $\lim_{j \rightarrow \infty} \rho_{j+1}/\rho_j = 4\rho/(1+\rho)^2 < 1$ , et

$$\sigma_\infty^2 = \frac{\rho(\rho^3 - 4\rho^2 + 5\rho + 2)}{\mu^2(1-\rho)^4}.$$

Pour  $\rho$  proche de 1,  $(\rho^3 - 4\rho^2 + 5\rho + 2) \approx 4$ , aussi  $\sigma_\infty^2 \approx 4\rho/[\mu^2(1-\rho)^4] \approx 4\sigma^2/(1-\rho)^2 \gg \sigma^2$ .

Par exemple, si  $\mu = 1$  et  $\rho = 0.9$ , alors  $\rho_{j+1}/\rho_j \rightarrow 3.6/3.61 \approx 0.9972$ ,  $\rho_{200} \approx 0.30$ ,  $\sigma^2 = 99$  and  $\sigma_\infty^2 = 39890$ .

Whitt (1989) donne des approximations de  $\sigma_\infty^2$  pour des quantités autres que  $W_j$ , et pour des files  $M/G/1$ .

Tout ceci peut aussi permettre d'estimer le temps requis par une simulation pour obtenir un intervalle de confiance d'une largeur donnée.

### 6.10.1 Détection et réduction du biais initial

Idéalement, nous souhaiterions générer l'état initial selon la loi d'équilibre, En effet, dans un tel cas, il n'y aurait aucun biais, mais une telle manière de procéder

est en général très délicate. Pratiquement, il n'est dès lors pas possible de procéder ainsi, et il nous faudra considérer le biais résultant de l'état initial de la simulation. Si  $(\mu_i, \sigma_i^2, \rho_{i,i+j}) \rightarrow (\mu, \sigma^2, \rho_j)$  quand  $i \rightarrow \infty$ , la question est alors de savoir comment détecter et réduire le biais  $|E[\bar{C}_n] - \mu|$  ? Les solutions pratiques sont avant tout heuristiques, bien que la méthode régénérative, expliquée à la Section 6.11.1, contourne purement et simplement le problème.

Considérons un processus en temps discret  $\{C_i, i \geq 1\}$  (le cas continu est similaire), et supposons que l'on ne compte pas les  $n_0$  premières observations :

$$\bar{C}_{n_0, n} = \frac{1}{n - n_0} \sum_{j=n_0+1}^n C_j.$$

Souvent, nous pouvons écrire  $|E[C_i] - \mu| \leq \kappa_0 \beta^i$  (i.e.  $|E[C_i] - \mu| \in O(\beta^i)$ ) pour  $\kappa_0 < \infty$  et  $\beta < 1$ . Dans ce cas, nous avons

$$\begin{aligned} |E[\bar{C}_{n_0, n}] - \mu| &\leq \frac{\kappa_0}{n - n_0} \sum_{i=n_0+1}^n \beta^i = \frac{\kappa_0 \beta^{n_0+1} (1 - \beta^{n-n_0})}{(n - n_0)(1 - \beta)} \\ &\in O\left(\frac{\beta^{n_0+1}}{(n - n_0)(1 - \beta)}\right) \approx O\left(\frac{\beta^{n_0}}{n(1 - \beta)}\right) \end{aligned}$$

si  $n$  est très grand par rapport à  $n_0$ . En d'autres termes,  $|E[\bar{C}_n] - \mu|$  est en  $O(1/(n(1 - \beta)))$ , si  $n \gg n_0$ .

En augmentant  $n_0$ , nous diminuons le biais, mais ce faisant, nous augmentons aussi la variance, qui est dans  $O(1/(n - n_0))$ . Quand  $n$  tend vers l'infini, le carré du biais devient négligeable par rapport à la variance, ainsi que, par conséquent, sa contribution dans le MSE. Il n'existe cependant aucune méthode fiable et universelle permettant de choisir  $n_0$  (ou  $t_0$ ) en pratique. Une manière populaire de procéder est d'utiliser l'heuristique de Welch, que nous pouvons décrire comme suit.

1. Faire  $k$  simulations de longueur  $n_1$ . Soit  $C_{ij}$  l'observation  $j$  de la répétition  $i$ .
2. Posons  $\bar{C}_j = \sum_{i=1}^k C_{ij}/k$ .
3. Lissage des hautes fréquences par une moyenne mobile de largeur  $w$  :

$$\bar{C}_j(w) = \begin{cases} \frac{1}{2w+1} \sum_{s=-w}^w \bar{C}_{j+s}, & j = w + 1, \dots, n_1 - w, \\ \frac{1}{2j-1} \sum_{s=-(j-1)}^{j-1} \bar{C}_{j+s}, & j = 1, \dots, w. \end{cases}$$

4. Regardons le graphique de  $\bar{C}_j(w)$  en fonction de  $j$  et soit  $n_0$  la valeur de  $j$  où le processus semble stable (choix *subjectif*).

Schématiquement, la procédure peut être résumée comme dans le Tableau 6.2.

Simulation 1	$C_{11}, C_{12}, \dots, C_{1,j-w}, \dots, C_{1,j+w}, \dots, C_{1,T_1}$
Simulation 2	$C_{21}, C_{22}, \dots, C_{2,j-w}, \dots, C_{2,j+w}, \dots, C_{2,T_1}$
$\vdots$	
Simulation $N$	$C_{N1}, C_{N2}, \dots, C_{N,j-w}, \dots, C_{N,j+w}, \dots, C_{N,T_1}$
Moyenne	$\bar{C}_1, \bar{C}_2, \dots, \bar{C}_{j-w}, \dots, \bar{C}_{j+w}, \dots, \bar{C}_{T_1}$
Moyenne mobile	$\bar{C}_{j-w}(w)$

TABLE 6.2 – Heuristique de Welch

## 6.11 Horizon tronqué

Une fois  $n_0$  (ou  $t_0$ ) choisi, nous faisons  $k$  répétitions indépendantes de longueurs  $T_1, \dots, T_k$ , où les  $T_i$  sont des variables déterministes ou aléatoires. Pour la répétition  $i$ , posons

$$X_i = \frac{1}{T_i - t_0} \sum_{j=N(t_0)+1}^{N(T_i)} C_j.$$

Un estimateur global est alors  $\bar{X}_k = (1/k) \sum_{i=1}^k X_i$ .

Si  $k$  est strictement plus grand que 2, nous pouvons exploiter le fait que les  $X_i$  sont i.i.d. pour estimer la variance et calculer un intervalle de confiance. Il s'agit l'approche "replication-deletion."

Dans le cas discret, nous avons

$$X_i = \frac{1}{n - n_0} \sum_{j=n_0+1}^n C_j.$$

Pour un budget de calcul fixe  $kn$ , comment choisir  $k$ ? Typiquement,  $k = 1$  minimise le MSE, mais pas toujours, et il est plus difficile d'estimer la variance lorsque  $k = 1$ .

Le MSE est minimisé pour  $k$  strictement plus grand que 1 lorsque

- (a) les autocorrélations diminuent très lentement et
- (b) le biais initial est très faible ou diminue très vite.

**Exemple 33.** Supposons que  $|E[C_j] - \mu| = \kappa_0 \beta^j$  et  $Cov[C_i, C_{i+j}] = \sigma^2 \rho_j = \sigma^2 \alpha^j$  pour  $j \geq 0$ , où  $\beta < 1$ ,  $0 < \alpha < 1$ ,  $\kappa_0 > 0$ , et  $\sigma^2 > 0$ . Nous disposons d'un

budget de calcul  $B = nk$ . On veut choisir  $k$  et  $n_0$  pour minimiser

$$\begin{aligned} \text{MSE}[\bar{X}_k] &= (E[X_i - \mu])^2 + \text{Var}[X_i]/k \\ &= \left( \frac{\kappa_0(\beta^{n_0+1} - \beta^{n+1})}{(n - n_0)(1 - \beta)} \right)^2 + \frac{\sigma^2}{(n - n_0)k} \left( 1 + 2 \sum_{j=1}^{n-n_0-1} \frac{\alpha^j(n - n_0 - j)}{n - n_0} \right), \end{aligned}$$

où nous avons utilisé les résultats de la Section 6.10 pour dériver l'égalité. Il faut prendre  $k > 1$  si  $\beta$  et  $\kappa_0$  sont petits et si  $\sigma^2$  et  $\alpha$  sont grands.

Mais si  $n_0$  et  $n$  sont fixés et  $k \rightarrow \infty$ , le biais ne disparaît pas et le TLC devient

$$\frac{\sqrt{k}(\bar{X}_k - \mu)}{\sigma_{n_0,n}} \Rightarrow N(0, 1) + \frac{\sqrt{k}\beta_{n_0,n}}{\sigma_{n_0,n}},$$

où  $\beta_{n_0,n} = E[X_i] - \mu$  et  $\sigma_{n_0,n}^2 = \text{Var}[X_i]$ . Un intervalle de confiance basé sur l'hypothèse que les  $X_i$  sont i.i.d. normales est asymptotiquement valide seulement si  $\sqrt{k}\beta_{n_0,n}/\sigma_{n_0,n} \rightarrow 0$  quand  $k \rightarrow \infty$ . Autrement dit, le biais doit converger vers zéro plus vite que l'écart-type de  $\bar{X}_k$ .

Si  $|E[C_j] - \mu| = \kappa_0\beta^j$  et  $\rho_j = \sigma^2\alpha^j$ , alors

$$\beta_{n_0,n} = \frac{\kappa_0(\beta^{n_0+1} - \beta^{n+1})}{(n - n_0)(1 - \beta)} \in O\left(\frac{\beta^{n_0}}{n - n_0}\right)$$

et

$$\frac{\sigma_{n_0,n}^2}{k} = \frac{\sigma^2}{(n - n_0)k} \left( 1 + 2 \sum_{j=1}^{n-n_0-1} \frac{\alpha^j(n - n_0 - j)}{n - n_0} \right) \in O\left(\frac{1}{(n - n_0)k}\right).$$

Ainsi,

$$\frac{\sqrt{k}\beta_{n_0,n}}{\sigma_{n_0,n}} \in O\left(\sqrt{k/(n - n_0)}\beta^{n_0}\right).$$

Cette expression converge vers 0 lorsque  $k \rightarrow \infty$  si et seulement si

$$\ln\left(\sqrt{\frac{k}{n - n_0}}\beta^{n_0}\right) \rightarrow -\infty$$

quand  $k$  tend vers l'infini, ou, en d'autres termes, si et seulement si  $\ln[k/(n - n_0)]/2 + n_0 \ln \beta \rightarrow -\infty$ . Comme  $\beta$  est supposé strictement plus petit que 1, cela revient à exiger

$$\frac{\ln[k/(n - n_0)]}{2 \ln \beta} + n_0 \rightarrow \infty.$$

Autrement dit, si on augmente  $k$ , il faut aussi augmenter  $n$  et  $n_0$  assez vite.

### 6.11.1 Intervalle basé sur une simulation unique

Supposons ici que le processus  $\{C_j, j \geq 1\}$  est stationnaire, avec  $E[C_j] = \mu$ , (e.g., la partie “échauffement” a déjà été enlevée.) Si nous estimons  $\mu$  par  $\bar{C}_n$ , comment pouvons-nous estimer  $\text{Var}[\bar{C}_n]$ ? Nous allons considérer deux techniques particulières : les moyennes par lots et la simulation régénération. Pour plus de détails techniques, le lecteur est renvoyé à la Section 9.5.3 de Law [8].

#### Moyennes par lots (“batch means”)

C’est l’approche la plus simple et la plus populaire pour les systèmes complexes. L’idée de base consiste à regrouper les  $n$  observations en  $k$  lots de taille  $\ell = n/k$ . Nous pouvons alors écrire

– moyenne pour le lot  $i$  :

$$X_i = \frac{1}{\ell} \sum_{j=\ell(i-1)+1}^{\ell i} C_j.$$

– moyenne globale :

$$\bar{X}_k = \frac{1}{k} \sum_{i=1}^k X_i = \bar{C}_n.$$

Si  $\ell$  est grand, on s’attend à ce que les  $X_i$  soient très peu corrélés et suivent à peu près la loi normale. Sous certaines conditions additionnelles, il est possible de prouver que c’est le cas pour  $\ell \rightarrow \infty$ , et  $k$  fixé. Nous pouvons alors calculer un intervalle de confiance en faisant l’hypothèse que les  $X_i$  sont i.i.d.  $N(\mu, \sigma_x^2)$ , et que donc  $\sqrt{k}(\bar{X}_k - \mu)/S_k \sim \text{Student}(k-1)$ . Il est souvent recommandé de choisir  $\ell$  le plus grand possible et  $k \leq 30$ . Un  $k$  plus petit diminue le *bias*  $E[S_k^2/k] - \text{Var}[\bar{X}_k]$ , mais augmente la variance de  $S_k^2/k$ .

#### Simulation régénérative

L’idée consiste à écrire la moyenne sur horizon infini comme un rapport de deux espérances sur horizon fini. Le processus doit cependant répondre à la définition de processus régénératif, comme donnée ci-dessous.

**Definition 6.3:** Processus régénératif

Un processus stochastique  $\{Y(t), t \geq 0\}$  est *régénératif* (au sens classique) s’il

existe une variable aléatoire  $\tau_1 > 0$  telle que  $\{Y(\tau_1 + t), t \geq 0\}$  est stochastiquement équivalent à  $\{Y(t), t \geq 0\}$  et indépendant de  $\tau_1$  et de  $\{Y(t), t < \tau_1\}$ .

La variable aléatoire  $\tau_1$  est un *instant de régénération*. La trajectoire du processus sur l'intervalle de temps  $(0, \tau_1]$  est un *cycle régénératif*, et si  $E[\tau_1] < \infty$ , le processus est dit *récurrent positif*.

L'adaptation au cas discret est évidente.

Si  $\{Y(t), t \geq 0\}$  est régénératif, alors  $\{Y(\tau_1 + t), t \geq 0\}$  est aussi régénératif avec un instant de régénération  $\tau_2$ , etc. On a ainsi une suite infinie d'instant de régénération  $0 = \tau_0 < \tau_1 < \tau_2 \dots$  et de cycles régénératifs i.i.d. Parfois, le premier cycle est différent des autres : le processus est régénératif *avec délai* (à partir de  $\tau_1$ ), mais l'effet est négligeable à long terme. Si  $\{Y(t), t \geq 0\}$  est régénératif aux instants  $\tau_1, \tau_2, \dots$  et si  $C(t) = f(Y(t))$ , alors  $\{C(t), t \geq 0\}$  est aussi régénératif aux mêmes instants.

**Exemple 34.** *File GI/G/1 stable.*

Si  $W_1 = 0$ , si les v.a.  $S_i - A_i$  sont i.i.d. et  $E[S_i - A_i] < 0$ , alors  $\{W_i, i \geq 1\}$  est régénératif et on peut prendre pour instants de régénération les époques  $i$  où  $W_i = 0$ . Le processus  $\{Q(t), t \geq 0\}$  régénère aussi, aux instants où un client arrive dans un système vide.

Mais peut-on prendre les instants où le système se vide comme instants de régénération ? La réponse est positive seulement dans le cas M/G/1.

**Exemple 35** (Centre d'appels). *Supposons qu'un centre d'appels opère pour une suite infinie de jours i.i.d.. Il ouvre à 8h et ferme à 21h.*

Soit  $t$  le temps écoulé depuis le premier jour à minuit, en heures, et  $Q(t) =$  le nombre d'appels dans la file au temps  $t$ . Le processus  $\{Q(t), t \geq 0\}$  régénère à  $\tau_j = 24j$  pour  $j = 1, 2, \dots$

Si  $X_j =$  nombre d'appels reçus au jour  $j$ ,  $\{X_j, j \geq 0\}$  régénère à  $\tau_j = j$  pour tout  $j$  (processus de renouvellement). Il en est de même pour  $\{Z_j, j \geq 0\}$  si  $Z_j =$  nombre d'abandons au jour  $j$ .

**Théorème du renouvellement avec gains**

Soit  $\{C_i, i \geq 0\}$  un processus régénératif aux instants  $\tau_j, j \geq 0$ , et  $N(t)$  le nombre d'événements qui se sont produits au cours de l'intervalle  $[0, t]$ . Posons

$$X_j = V_{N(\tau_j)} - V_{N(\tau_{j-1})},$$

le coût pour le cycle  $j$ , et

$$Y_j = \tau_j - \tau_{j-1},$$

la durée du cycle  $j$ .

**Théorème 7.** Si  $E[Y_j] > 0$  et  $E[|X_j|] < \infty$ , alors,

$$\bar{v} \stackrel{\text{def}}{=} \lim_{t \rightarrow \infty} \frac{E[V_{N(t)}]}{t} = \frac{E[X_j]}{E[Y_j]} \quad (\text{version espérance}),$$

et

$$\bar{v} \stackrel{\text{p.s.}}{=} \lim_{t \rightarrow \infty} \frac{V_{N(t)}}{t} \quad (\text{version trajectoire}).$$

Une fois le nombre de cycles  $n$  fixé, nous retrouvons le problème d'estimation du quotient  $\bar{v} = E[X_j]/E[Y_j]$  à partir des observations i.i.d.  $(X_1, Y_1), \dots, (X_n, Y_n)$ , comme étudié au sein de la Section 6.9. Toutefois, pour une durée de la simulation à  $t$ , le nombre de cycles  $M(t) = \sup\{n \geq 0 : \tau_n \leq t\}$  est aléatoire. Lorsque nous atteignons  $t$ , nous pouvons terminer le cycle en cours (nous aura  $M(t) + 1$  cycles), ou ignorer le cycle en cours (nous en aurons  $M(t)$ ). La variance est dans  $O(1/t)$  dans les deux cas. Le biais sur  $\bar{v}$  est dans  $O(1/t^2)$  pour la première approche, et  $O(1/t)$  pour la seconde.

**Théorème 8.** Sous les conditions du théorème de la limite centrale pour un quotient, quand  $t \rightarrow \infty$ ,

$$\begin{aligned} \frac{\sqrt{M(t)}(\hat{v}_{M(t)} - v)}{\hat{\sigma}_{g,M(t)}} &\Rightarrow \frac{\sqrt{t/\bar{Y}_{M(t)}}(\hat{v}_{M(t)} - v)}{\hat{\sigma}_{g,M(t)}} \\ &\Rightarrow \frac{\sqrt{t/E[Y_1]}(\hat{v}_{M(t)} - v)}{\hat{\sigma}_{g,M(t)}} \\ &\Rightarrow N(0, 1), \end{aligned}$$

où  $\sigma_g$  est défini comme en (6.3). (??) est valide aussi si nous remplaçons  $M(t)$  par  $M(t) + 1$ .



### 6.11.2 Coûts (ou revenus) actualisés

Considérons un taux d'actualisation  $\rho > 0$ . Pour un état initial  $S_0$ , nous voulons estimer

$$v_\rho^\infty = \sum_{i=1}^{\infty} E[e^{-\rho t_i} C_i] = E[V_\rho^\infty]$$

où  $\{C_i, i \geq 0\}$  est supposé être un processus *régénératif* et

$$\begin{aligned} V_\rho^\infty &= \sum_{i=1}^{\infty} e^{-\rho t_i} C_i \\ &= V_{\rho, N(\tau_1)} + \sum_{i=N(\tau_1)+1}^{\infty} e^{-\rho t_i} C_i \\ &= V_{\rho, N(\tau_1)} + e^{-\rho \tau_1} \sum_{i=N(\tau_1)+1}^{\infty} e^{-\rho(t_i - \tau_1)} C_i. \end{aligned}$$

En prenant l'espérance de cette dernière expression, nous obtenons

$$E[V_\rho^\infty] = E[V_{\rho, N(\tau_1)}] + E[e^{-\rho \tau_1}] E[V_\rho^\infty]$$

et donc

$$v_\rho^\infty = E[V_\rho^\infty] = \frac{E[V_{\rho, N(\tau_1)}]}{1 - E[e^{-\rho \tau_1}]}.$$

Une fois encore, nous sommes ramené à estimer un quotient de deux espérances sur horizon fini.

## 6.12 Résumé du chapitre

La construction d'un intervalle de confiance permet de mesurer la qualité statistique d'un modèle, mais pas la pertinence du modèle par rapport à la réalité qu'il vise à représenter. La composante clé dans la construction d'un intervalle de confiance est le théorème de la limite centrale, et son extension, le théorème delta, appliqué notamment dans le cas important de l'estimation d'un rapport d'espérances puisque ce type d'estimateur apparaît dans plusieurs, dont certains problèmes d'estimation sur un horizon infini. Il ne faut néanmoins pas perdre de vue les conditions d'applications de ces théorèmes et leur caractère asymptotique. Outre le théorème de la limite centrale, la méthode "bootstrap" représente un moyen efficace et facile à mettre en oeuvre pour évaluer les propriétés statistiques d'un modèle, et permet de ne pas faire d'hypothèse sur les distributions sous-jacentes. L'idée principale consiste à tirer avec remplacement dans l'échantillon original. Dans le cas de processus stochastiques avec un horizon infini, les choses se compliquent un peu, et on ne s'attarde

ici que sur les processus stationnaires. L'état stationnaire est rarement atteint dès le début de la simulation, d'où le besoin d'une phase de réchauffement en début de simulation afin d'éliminer le biais initial. A budget de calcul égal, il est souvent préférable de faire une seule longue simulation plutôt que plusieurs répétitions en raison du biais qui peut alors devenir prédominant. Cependant, une seule simulation pose la question de l'estimation de variance. Pour y répondre, on pourra procéder par l'utilisation de moyennes par lots, ou, dans le cadre de processus régénératifs, de cycles régénératifs.

### 6.13 Exercices

1. Nous souhaitons estimer l'espérance du coût moyen par jour d'un certain système, en supposant que les jours sont indépendants. Nous avons déjà simulé  $n = 1000$  et avons obtenu une moyenne échantillonnage  $\bar{X}_n$  égale à 1073.5 et une variance empirique  $S_n^2$  de 5890. Combien d'exécutions additionnelles de simulations devrions-nous faire si nous voulons un intervalle de confiance à 95% de demi-largeur au plus égal à 2 ?
2. Considérons un estimateur d'une certaine quantité  $\mu$  (on ne précise pas ici ce que représente  $\mu$ ). Réduire la variance de cet estimateur garantit-il qu'il sera plus efficace ? En quoi une technique de réduction de variance est-elle habituellement considérée comme une technique d'amélioration d'efficacité ?
3. On considère une file  $M/M/1$ . Nous souhaitons estimer le temps d'attente moyen par client

$$w = \lim_{T \rightarrow \infty} \frac{E[W(0, T)]}{E[X(0, T)]},$$

où  $W(t_1, t_2)$  est la somme des temps d'attente pendant l'intervalle de temps  $[t_1, t_2]$  et  $X(t_1, t_2)$  est le nombre de clients servis pendant ce même intervalle. Pour estimer  $w$ , nous allons tester deux approches : tronquer l'horizon et simuler des répliques indépendantes, simuler une réplique et la diviser en lots de taille égale. Sous l'hypothèse de normalité, on souhaite également établir un intervalle de confiance à 95% pour chacune de ces approches.

On supposera dans cet exemple que les clients arrivent selon un processus de Poisson de taux  $\lambda = 1$  et sont servis par un seul serveur pendant une durée exponentielle de moyenne  $1/\mu = 1/2$ . A titre de simplification, on supposera qu'aucun client n'abandonne. On procédera tout d'abord en effectuant  $n = 30$  répliques indépendantes sur un horizon tronqué de durée  $T = 10000$ . Nous utiliserons ensuite 30 lots de durée 10000 après

une période de réchauffement de 1000. On considérera une approche par événements dans l'implémentation.

Comparez les résultats obtenus.

4. Quelles sont les conditions nécessaires pour parler de système régénératif ? Sont-elles suffisantes pour pouvoir appliquer la technique des cycles régénératifs ? Est-il toujours conseillé d'utiliser cette technique plutôt que l'approche par lots pour un système régénératif ? Pourquoi ?



## Chapitre 7

# Amélioration de l'Efficacité

### 7.1 Exemple introductif

Rappelons que l'efficacité  $\text{Eff}[X]$  d'un estimateur  $X$  se définit comme suit :

$$\text{Eff}[X] = \frac{1}{\text{MSE}[X]C(X)}.$$

Est-il possible, étant donné  $X$ , de construire un nouvel estimateur  $Y$  plus efficace que  $X$  ? Nous allons introduire les principaux concepts d'amélioration de l'efficacité au moyen d'un exemple introductif sur les centres d'appels téléphoniques.

Posons  $B$ , le *facteur d'achalandage* pour la journée, et supposons que  $P[B = b_t] = q_t$ , où

$t$	1	2	3	4
$b_t$	0.8	1.0	1.2	1.4
$q_t$	0.25	0.55	0.15	0.05

Il est facile de vérifier que  $E[B] = 1$ . Les *arrivées* des appels suivent processus de Poisson de taux  $B\lambda_j$  durant l'heure  $j$ . Notons  $G_i(s)$  = nombre d'appels dont le service a débuté après moins de  $s$  secondes d'attente, le jour  $i$ . On veut estimer  $\mu = E[G_i(s)]$ , disons pour  $s = 20$ . Les données du problème sont résumées dans la Table 7.1. Nous supposons de plus que les durées de service des appels suivent la loi Gamma( $\alpha, \gamma$ ), dont la moyenne est  $\alpha\gamma$ . Dans notre exemple, on a  $\alpha = 1$  et  $\gamma = \gamma_1 = 100$ .

TABLE 7.1 – Nombre d'agents  $n_j$  et taux d'arrivée  $\lambda_j$  (par heure) pour 13 périodes d'une heure dans le centre d'appel.

$j$	0	1	2	3	4	5	6	7	8	9	10	11	12
$n_j$	4	6	8	8	8	7	8	8	6	6	4	4	4
$\lambda_j$	100	150	150	180	200	150	150	150	120	100	80	70	60

On simule  $n$  jours, indépendamment. Soit  $X_i = G_i(s)$  pour le jour  $i$ , et

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i.$$

On a  $E[\bar{X}_n] = \mu$  et  $\text{Var}[\bar{X}_n] = \text{Var}[X_i]/n$ .

Une expérience avec  $n = 1000$  donne  $\bar{X}_n = 1518.3$  et  $S_n^2 = 21615$ . La variance estimée de  $\bar{X}_n$  est alors  $\widehat{\text{Var}}[\bar{X}_n] = 21.6$ .

Nous souhaitons construire un intervalle de confiance pour  $\sigma^2 = n\text{Var}[\bar{X}_n]$ , sous l'hypothèse que  $(n-1)S_n^2/\sigma^2$  suit approximativement une  $\chi^2$  à  $n-1$  degrés de liberté. Ceci permet de construire l'intervalle à 90% :  $[0.930S_n^2, 1.077S_n^2]$ . En d'autres termes, l'erreur relative de cet estimateur est inférieure à 8% avec une "confiance" d'environ 90%. Voyons comment *améliorer* cet estimateur  $\bar{X}_n$ , en réduisant sa variance. Pour chaque méthode proposée, nous donnerons des résultats numériques pour  $n = 1000$ .

### 7.1.1 Estimation indirecte.

Soit  $A_i$  le nombre total d'arrivées au jour  $i$ ; posons  $D_i = A_i - X_i$ . On sait que  $a = E[A_i] = \sum_{j=1}^m \lambda_j = 1660$ . Nous pouvons dès lors écrire  $\mu = E[X_i] = E[A_i - D_i] = a - E[D_i]$ , que l'on peut estimer par

$$\bar{X}_{i,n} = E[A_i] - \bar{D}_n = a - \frac{1}{n} \sum_{i=1}^n D_i.$$

Cet estimateur a moins de variance que  $\bar{X}_n$  ssi  $\text{Var}[D_i] < \text{Var}[X_i]$ . Variance estimée :  $\widehat{\text{Var}}[X_{i,i}] = 18389$ .

### 7.1.2 Variable de contrôle (VC).

Idée : exploiter l'information auxiliaire.

Par exemple, si  $A_i$  est plus grand que d'habitude ( $A_i > E[A_i] = 1660$ ), on s'attend à ce que ce jour là,  $X_i$  et  $D_i$  surestiment  $E[X_i]$  et  $E[D_i]$ .

On pourrait faire une "correction" à ces estimateurs : remplacer  $X_i$  par

$$X_{c,i} = X_i - \beta(A_i - 1660)$$

où  $\beta$  est une constante appropriée. Alors

$$\bar{X}_{c,n} = \bar{X}_n - \beta(\bar{A}_n - 1660).$$

On a  $E[\bar{X}_{c,n}] = E[X_i]$  et

$$\text{Var}[\bar{X}_{c,n}] = \frac{\text{Var}[X_i] + \beta^2 \text{Var}[A_i] - 2\beta \text{Cov}[A_i, X_i]}{n}.$$

Cette variance est une fonction quadratique en  $\beta$ , que l'on minimise en prenant

$$\beta = \beta^* = \text{Cov}[A_i, X_i] / \text{Var}[A_i].$$

La variance minimale est

$$\text{Var}[\bar{X}_{c,n}] = \frac{\text{Var}[X_i] - (\beta^*)^2 \text{Var}[A_i]}{n} = \text{Var}[\bar{X}_n] (1 - \rho^2[A_i, X_i])$$

où  $\rho[A_i, X_i]$  est le coeff. de corrélation entre  $A_i$  et  $X_i$ .

On ne connaît pas  $\text{Cov}[A_i, X_i]$ , mais :

- (a) On peut l'estimer par des *expériences pilotes*.
- (b) On peut l'estimer par les *mêmes  $n$  simulations* que  $\bar{X}_n$ .

Avec (b) on obtient l'estimateur (légèrement biaisé) :

$$\bar{X}_{ce,n} = \bar{X}_n - \frac{1}{n-1} \left[ \sum_{i=1}^n (A_i - \bar{A}_n)(X_i - \bar{X}_n) \right] \frac{\bar{A}_n - a}{\text{Var}[A_i]}.$$

Conditionnellement à  $B_i$ ,  $A_i \sim \text{Poisson}(1660B_i)$ . On a donc

$$\begin{aligned} \text{Var}[A_i] &= \text{Var}[E[A_i|B_i]] + E[\text{Var}[A_i|B_i]] = \text{Var}[1660B_i] + E[1660B_i] \\ &= 1660^2 \text{Var}[B_i] + 1660E[B_i] = 67794.4. \end{aligned}$$

Variance empirique obtenue ici : 3310.

En prenant  $\beta = 1$ , on retrouve l'estimateur indirect :

$$\bar{X}_{i,n} = \bar{X}_n - (\bar{A}_n - 1660) = 1660 - \bar{D}_n.$$

Si on combine *VC* + *indirect*, on obtient :

$$\begin{aligned}\bar{X}_{i,c,n} &= a - \bar{D}_n - \beta_2(\bar{A}_n - a) \\ &= \bar{A}_n - \bar{D}_n - (1 + \beta_2)(\bar{A}_n - a) \\ &= \bar{X}_n - (1 + \beta_2)(\bar{A}_n - a),\end{aligned}$$

i.e.,  $\bar{X}_{i,c,n}$  est équivalent à  $\bar{X}_{c,n}$  avec  $\beta = 1 + \beta_2$ . Par conséquent, en présence de la variable de contrôle, l'estimation indirecte n'apporte rien.

Nous pourrions aussi considérer d'autres variables de contrôle, comme  $B_i$ , la moyenne des durées de service, etc.

### 7.1.3 Stratification.

Le facteur d'achalandage  $B_i$  est une source importante de variabilité importante dans le cas présent. Essayons de la contrôler. En posant  $\mu_t = E[X_i | B_i = b_t]$ , on a

$$\begin{aligned}\mu &= E[X_i] = \sum_{t=1}^4 P[B_i = b_t] \cdot E[X_i | B_i = b_t] \\ &= .25 E[X_i | B_i = 0.8] + .55 E[X_i | B_i = 1.0] \\ &\quad + .15 E[X_i | B_i = 1.2] + .05 E[X_i | B_i = 1.4] \\ &= .25 \mu_1 + .55 \mu_2 + .15 \mu_3 + .05 \mu_4.\end{aligned}$$

*Idée* : estimer  $\mu_t$  *séparément* pour chaque  $t$ . Supposons qu'il y a  $N_t$  jours où  $B_i = b_t$  et soient  $X_{t,1}, \dots, X_{t,N_t}$  les valeurs de  $X_i$  pour ces jours. On peut estimer  $\mu_t = E[X_i | B_i = b_t]$  par

$$\hat{\mu}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} X_{t,i}$$

et  $\mu$  par

$$\bar{X}_{s,n} = \sum_{t=1}^4 q_t \hat{\mu}_t = .25 \hat{\mu}_1 + .55 \hat{\mu}_2 + .15 \hat{\mu}_3 + .05 \hat{\mu}_4.$$

On a

$$\begin{aligned}\text{Var}[\bar{X}_{s,n} \mid N_1, N_2, N_3, N_4] &= \sum_{t=1}^4 q_t^2 \text{Var}[\hat{\mu}_t \mid N_t] = \sum_{t=1}^4 q_t^2 \sigma_t^2 / N_t \\ &= .25^2 \sigma_1^2 / N_1 + .55^2 \sigma_2^2 / N_2 + .15^2 \sigma_3^2 / N_3 + .05^2 \sigma_4^2 / N_4.\end{aligned}$$

où  $\sigma_t^2 = \text{Var}[X_i \mid B_i = b_t]$ .

La variance est réduite si les  $\sigma_t^2$  sont inférieurs à  $\text{Var}[X_i]$ .

Si les  $B_i$  sont générés normalement : *post-stratification*. Pour estimer  $\mu$  par stratification, on peut aussi *fixer les*  $N_t = n_t$  *à l'avance*, c'est-à-dire choisir à l'avance combien de jours on aura  $B_i = b_t$  pour chaque valeur de  $t$ .

– *Allocation proportionnelle* : prendre  $n_t = n q_t$ .

Avec  $n = 1000$ , cela donne  $n_1 = 250$ ,  $n_2 = 550$ ,  $n_3 = 150$ ,  $n_4 = 50$ .

– *Allocation optimale* : choisir les  $n_t$  pour minimiser  $\text{Var}[\bar{X}_{s,n}]$  sous la contrainte  $n_1 + n_2 + n_3 + n_4 = n$ . On obtient :

$$\frac{n_t}{n} = \frac{\sigma_t P[B_i = t]}{\sum_{k=1}^4 \sigma_k P[B_i = k]} = \frac{\sigma_k q_k}{\sum_{k=1}^4 \sigma_k q_k}.$$

On ne connaît pas ces  $\sigma_k$ , mais on peut les estimer par des essais pilotes.

Avec  $n_0 = 800$  essais pilotes, 200 par valeur de  $t$ , on obtient par exemple  $(n_1, n_2, n_3, n_4) = (219, 512, 182, 87)$  (après arrondi).

### 7.1.4 Stratégies combinées.

*Stratification combinée avec VC :*

$$\hat{\mu}_t = \frac{1}{n_t} \sum_{i=1}^{n_t} X_{c,t,i} = \frac{1}{n_t} \sum_{i=1}^{n_t} [X_{t,i} - \beta_t (A_{t,i} - a b_t)].$$

On minimise  $\sigma_t^2 = \text{Var}[X_{c,t,i}]$  en prenant

$$\beta_t = \beta_t^* = \text{Cov}[A_{t,i}, X_{t,i}] / \text{Var}[A_{t,i}] = \text{Cov}[A_i, X_i \mid B_i = b_t] / (a b_t).$$

L'ajout d'une VC change les  $\sigma_t^2$  : l'allocation optimale n'est plus la même. Avec  $n_0 = 800$  essais pilotes, on obtient  $(\beta_1, \beta_2, \beta_3, \beta_4) = (1.020, 0.648, 0.224, -0.202)$  et  $(n_1, n_2, n_3, n_4) = (131, 503, 247, 119)$  comme estimation des valeurs optimales.

En répétant l'expérience avec  $n = 100000$ , on peut trouver les estimations suivantes pour la variance ( $\pm 1\%$ ) :

$$\text{Var}[X_n] = 21998 ; \text{Var}[X_{i,n}] = 17996 ; \text{Var}[X_{c,n}] = 3043 ; \text{Var}[X_{so,c,n}] = 885.$$

### 7.1.5 Résultats numériques pour $n = 1000$

Méthode	Estimateur	Mean	$S_n^2(\pm 9\%)$	Ratio
Crude estimator	$\bar{X}_n$	1518.2	21615	1.000
Indirect	$\bar{X}_{i,n}$	1502.5	18389	0.851
CV $A_i$ , with pilot runs	$\bar{X}_{c,n}$	1510.1	3305	0.153
CV $A_i$ , no pilot runs	$\bar{X}_{ce,n}$	1510.2	3310	0.153
Indirect + CV, no pilot runs	$\bar{X}_{i,c,n}$	1510.1	3309	0.153
Stratification (propor.)	$\bar{X}_{sp,n}$	1509.5	1778	0.082
Stratification (optimal)	$\bar{X}_{so,n}$	1509.4	1568	0.073
Strat. (propor.) + CV	$\bar{X}_{sp,c,n}$	1509.2	1140	0.053
Strat. (optimal) + CV	$\bar{X}_{so,c,n}$	1508.3	900	0.042

### 7.1.6 Comparaison de deux systèmes similaires

Supposons à présent qu'il est possible de diminuer légèrement le paramètre  $\gamma$  à de la loi gouvernant les durées de service, en prenant

$$\gamma = \gamma_2 = \gamma_1(1 - \delta),$$

pour  $\delta > 0$  très petit. Si on utilise les mêmes nombres aléatoires, cela équivaut à multiplier les durées de service par  $(1 - \delta)$ .

Nous voulons estimer  $\mu(\gamma_2) - \mu(\gamma_1) = E_\gamma[X_2] - E_\gamma[X_1]$ , afin de mesurer l'effet d'augmenter un peu la vitesse des serveurs. On simule  $n$  jours pour chaque valeur de  $\gamma$ .

- $X_{1,i}$  = valeur de  $G_i(s)$  avec  $\gamma_1$  ;
- $X_{2,i}$  = valeur de  $G_i(s)$  avec  $\gamma_2$  ;
- $\Delta_i = X_{2,i} - X_{1,i}$ ,

$$\bar{\Delta}_n = \frac{1}{n} \sum_{i=1}^n \Delta_i$$

On peut simuler  $X_{1,i}$  et  $X_{2,i}$

- (i) avec des v.a. indépendantes (VAI),
- (ii) avec des v.a. communes (VAC).

Comme

$$\text{Var}[\Delta_i] = \text{Var}[X_{1,i}] + \text{Var}[X_{2,i}] - 2\text{Cov}[X_{1,i}, X_{2,i}],$$

le but des variables aléatoires communes est de rendre la covariance positive.

Comment implanter les VAC ? Utiliser des “random streams” différents pour générer :

1. le facteur d’achalandage  $B_i$  ;
2. les temps inter-arrivées ;
3. les durées des appels ;
4. les durées de patience.

Tout est généré par inversion : on utilise une uniforme par v.a.

### Synchronisation

Lorsqu’on change les durées de service, les durées d’attente changent et les décisions d’abandon peuvent ainsi changer. Si on ne génère les durées de service que pour les clients qui n’abandonnent pas, alors on peut perdre la synchronisation : on peut avoir une durée de service de moins à générer dans un système que dans l’autre.

On peut générer les durées de service :

- (a) pour tous les appels, même les abandons,
- (b) seulement pour les appels servis.

Le pour et le contre de chaque possibilité...

De même, la durée de patience n’a pas besoin d’être générée pour les clients qui n’attendent pas. On peut la générer :

- (c) pour tous les appels,
- (d) seulement si nécessaire.

Les résultats (avec  $n = 10^4$ ) sont résumés dans la Table 7.2.

## 7.2 Induction de corrélation

Conditions suffisantes pour que  $\text{Cov}[X, Y]$  soit positive, ou soit négative ?

Comment maximiser ou minimiser la corrélation pour des lois marginales données ?

**Theorème 9** (Bornes de Fréchet). *Parmi les paires de v.a.  $(X, Y)$  dont les f.r. marginales sont  $F$  et  $G$ , la paire  $(X, Y) = (F^{-1}(U), G^{-1}(U))$  où  $U \sim U(0, 1)$ , maximise  $\rho[X, Y]$ , et la paire  $(X, Y) = (F^{-1}(U), G^{-1}(1 - U))$  minimise  $\rho[X, Y]$ .*

Method	$\delta = 0.1$		$\delta = 0.01$		$\delta = 0.001$	
	$\bar{\Delta}_n$	$\widehat{\text{Var}}[\Delta_i]$	$\bar{\Delta}_n$	$\widehat{\text{Var}}[\Delta_i]$	$\bar{\Delta}_n$	$\widehat{\text{Var}}[\Delta_i]$
IRN (a + c)	55.2	56913	4.98	45164	0.66	44046
IRN (a + d)	52.2	54696	7.22	45192	-1.82	45022
IRN (b + c)	50.3	56919	9.98	44241	1.50	45383
IRN (b + d)	53.7	55222	5.82	44659	1.36	44493
CRN, no sync. (b + d)	56.0	3187	5.90	1204	0.19	726
CRN (a + c)	56.4	2154	6.29	37	0.62	1.8
CRN (a + d)	55.9	2161	6.08	158	0.74	53.8
CRN (b + c)	55.8	2333	6.25	104	0.63	7.9
CRN (b + d)	55.5	2323	6.44	143	0.59	35.3

TABLE 7.2 – Centre d'appel : tests sur les variables aléatoires communes

Par exemple, si la fonction de répartition d'une durée de service est  $F$  dans le premier système et  $G$  dans le second, et si on génère les durées de service par  $X = F^{-1}(U)$  et  $Y = G^{-1}(U)$ , alors  $\text{Cov}[X, Y] \geq 0$ .

**Theorème 10.** Soient  $X = f(U)$  où  $U \sim U(0, 1)$  et  $Y = g(V)$  où  $V \sim U(0, 1)$ . Alors,

- si  $f$  et  $g$  sont monotones dans le même sens et  $V = U$ , alors  $\text{Cov}[X, Y] \geq 0$  ;
- si  $f$  et  $g$  sont monotones dans le même sens et  $V = 1 - U$ , alors  $\text{Cov}[X, Y] \leq 0$ .

### 7.3 Valeurs aléatoires communes (VAC)

On utilise  $\Delta = X_2 - X_1$  pour estimer  $\mu_2 - \mu_1 = E[X_2] - E[X_1]$ . On a

$$\text{Var}[\Delta] = \text{Var}[X_1] + \text{Var}[X_2] - 2 \text{Cov}[X_1, X_2].$$

*Objectif* : induire une corrélation positive entre  $X_1$  et  $X_2$  sans changer leurs lois individuelles.

*Technique* : utiliser les mêmes nombres aléatoires pour simuler les deux systèmes, en essayant de maintenir la *synchronisation* le mieux possible.

Si  $X_k = f_k(\mathbf{U}_k) = f_k(U_{k,1}, U_{k,2}, \dots)$  pour  $k = 1, 2$ , utiliser des VAC partout veut dire prendre  $\mathbf{U}_1 = \mathbf{U}_2$ .

**Theorème 11.** Si  $f_1$  et  $f_2$  sont monotones dans le même sens par rapport à tous leurs arguments, alors  $\text{Cov}[X_1, X_2] \geq 0$ .

Pour maximiser la corrélation, il faudrait générer  $X_1$  et  $X_2$  directement par inversion !

**Exemple :** Processus de Lindley.

$W_{i+1} = \max(0, W_i + S_i - A_i)$ , où  $S_i - A_i$  est indépendant de  $W_i$ .

Si  $X_1$  et  $X_2$  sont des fonctions non-décroissantes des  $W_i$  pour deux processus de Lindley simulés avec des VAC, alors  $\text{Cov}[X_1, X_2] \geq 0$ .

Parfois il est très difficile de vérifier les conditions du théorème.

Par exemple, pour la banque ou un centre d'appels, la monotonie est difficile à vérifier à cause des possibilités d'abandon. De plus, les conditions du théorème sont *suffisantes*, mais *pas nécessaires*.

Pour tester si c'est efficace en pratique : faire un *expérience pilote avec les VAC* et estimer  $\text{Cov}(X_1, X_2)$ . Il n'est pas nécessaire de faire de simulations sans les VAC : pour estimer la variance qu'on aurait sans les VAC, il suffit de prendre la version empirique de  $\text{Var}[X_1] + \text{Var}[X_2]$ .

## 7.4 Variables de contrôle (VC)

*Idée :* exploiter de l'information auxiliaire pour faire une *correction* à l'estimateur. On se restreint ici aux VCs *linéaires*.

Soit  $X$  un estimateur sans biais de  $\mu$  et  $\mathbf{C} = (C^{(1)}, \dots, C^{(q)})^T$  des VCs corrélées avec  $X$ , d'espérance connue  $E[\mathbf{C}] = \boldsymbol{\nu} = (\nu^{(1)}, \dots, \nu^{(q)})^T$ .

L'estimateur avec VC est :

$$X_c = X - \boldsymbol{\beta}^T (\mathbf{C} - \boldsymbol{\nu}) = X - \sum_{\ell=1}^q \beta_{\ell} (C^{(\ell)} - \nu^{(\ell)}),$$

où  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_q)^T$  (des constantes).

On a  $E[X_c] = E[X] = \mu$ .

Comment choisir  $\boldsymbol{\beta}$  ?

Soient  $\boldsymbol{\Sigma}_c = \text{Cov}[\mathbf{C}]$  et  $\boldsymbol{\Sigma}_{cX} = (\text{Cov}(X, C^{(1)}), \dots, \text{Cov}(X, C^{(q)}))^T$ .

**Assumption 7.1 VC1**

$\text{Var}[X] = \sigma^2 < \infty$ ,  $\Sigma_c$  et  $\Sigma_{cX}$  sont finies, et  $\Sigma_c$  est définie positive (et donc inversible).

On a alors

$$\text{Var}[X_c] = \text{Var}[X] + \beta^T \Sigma_c \beta - 2\beta^T \Sigma_{cX}.$$

Pour minimiser par rapport à  $\beta$ , il suffit d'annuler le gradient par rapport à  $\beta$  :

$$0 = \nabla_{\beta} \text{Var}[X_c] = 2\Sigma_c \beta - 2\Sigma_{cX}.$$

Le minimum est donc atteint pour

$$\beta = \beta^* = \Sigma_c^{-1} \Sigma_{cX},$$

qui donne la variance minimale

$$\text{Var}[X_c] = (1 - R_{cX}^2) \text{Var}[X] \stackrel{\text{def}}{=} \sigma_c^2,$$

où

$$R_{cX}^2 = \frac{\Sigma_{cX}^T \Sigma_c^{-1} \Sigma_{cX}}{\text{Var}[X]}$$

(le carré du coefficient de corrélation multiple entre  $C$  et  $X$ ) et la variance est réduite par le facteur  $1 - R_{cX}^2 = \sigma_c^2 / \sigma^2$ . Mais avec  $\beta \neq \beta^*$ , la variance peut augmenter.

**7.4.1 Types de VCs**

- (a) variables *internes*, basées sur des quantités déjà calculées durant la simulation ;
- (b) variables *externes*, obtenues par des simulations additionnelles ;
- (c) VCs implicites obtenues via une *moyenne pondérée*.

Soient  $X^{(0)}, \dots, X^{(q)}$  des estimateurs sans biais de  $\mu$ . Posons

$$X_c = \sum_{\ell=0}^q \beta_{\ell} X^{(\ell)} = X^{(0)} - \sum_{\ell=1}^q \beta_{\ell} (X^{(0)} - X^{(\ell)})$$

où  $\sum_{\ell=0}^q \beta_{\ell} = 1$ .

On peut interpréter  $C^{(\ell)} = X^{(0)} - X^{(\ell)}$ ,  $\ell = 1, \dots, q$ , comme VC pour  $X = X^{(0)}$ .

### 7.4.2 Estimation de $\beta^*$ : propriétés asymptotiques

En pratique, on ne connaît pas  $\beta^* = \Sigma_c^{-1} \Sigma_{cX}$  (parfois  $\Sigma_c$ , mais jamais  $\Sigma_{cX}$ ). On peut l'estimer, disons par  $\hat{\beta}_n$ , calculé à partir de  $(X_1, C_1), \dots, (X_n, C_n)$ .

Posons

$$X_{ce,i} = X_i - \hat{\beta}_n^T (C_i - \nu),$$

et

$$\bar{X}_{ce,n} = \bar{X}_n - \hat{\beta}_n^T (\bar{C}_n - \nu).$$

**Théorème 12.** *Sous l'hypothèse CV1, lorsque  $n \rightarrow \infty$ , si  $\hat{\beta}_n \Rightarrow \beta^*$ , alors*

$$\begin{aligned} \sqrt{n}(\bar{X}_{c,n} - \bar{X}_{ce,n}) &\Rightarrow 0, \\ S_{ce,n}^2 &\stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n (X_{ce,i} - \bar{X}_{ce,n})^2 \Rightarrow \sigma_c^2, \\ \frac{\sqrt{n}(\bar{X}_{ce,n} - \mu)}{S_{ce,n}} &\Rightarrow \frac{\sqrt{n}(\bar{X}_{c,n} - \mu)}{\sigma_c} \Rightarrow N(0, 1). \end{aligned}$$

On peut utiliser ce théorème pour *calculer un IC* pour  $\mu$ , en supposant que  $\sqrt{n}(\bar{X}_{ce,n} - \mu)/S_{ce,n} \sim N(0, 1)$ .

Comment construire  $\hat{\beta}_n$ ? *Méthode de base :*

$$\hat{\beta}_n = \hat{\Sigma}_c^{-1} \hat{\Sigma}_{cX}$$

où les éléments de  $\hat{\Sigma}_c$  et  $\hat{\Sigma}_{cX}$  sont

$$\begin{aligned} \hat{\sigma}_c^{(\ell,k)} &= \frac{1}{n-1} \sum_{i=1}^n (C_i^{(\ell)} - \bar{C}_n^{(\ell)})(C_i^{(k)} - \bar{C}_n^{(k)}), \\ \hat{\sigma}_{cX}^{(\ell)} &= \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)(C_i^{(\ell)} - \bar{C}_n^{(\ell)}). \end{aligned}$$

### 7.4.3 Cas multinormal

Dans le cas où  $\begin{pmatrix} X_i \\ C_i \end{pmatrix} \sim \text{normal}$ , on peut utiliser la théorie de la *régression linéaire* (avec estimateurs moindres carrés) pour le modèle

$$X = \mu + \beta^T (C - \nu) + \epsilon$$

où  $\epsilon \sim N(0, \sigma_\epsilon^2)$ . Si on définit

$$\tilde{S}_{ce,n}^2 = \frac{n}{n-q-1} \left( \frac{1}{n} + \frac{(\bar{C}_n - \nu)^T \hat{\Sigma}_c^{-1} (\bar{C}_n - \nu)}{n-1} \right) \sum_{i=1}^n (X_{ce,i} - \bar{X}_{ce,n})^2,$$

où  $\hat{\beta}_n$  utilise les covariances empiriques, on a

**Theorème 13.** Si les  $(X_i, C_i^T)^T$  sont i.i.d normaux, alors

$$\begin{aligned} E[\bar{X}_{ce,n}] &= \mu \quad (\text{aucun biais}), \\ E[\tilde{S}_{ce,n}^2/n] &= \text{Var}[\bar{X}_{ce,n}] = \frac{n-2}{n-q-2} (1 - R_{cX}^2) \text{Var}[\bar{X}_n], \\ \sqrt{n}(\bar{X}_{ce,n} - \mu) / \tilde{S}_{ce,n} &\sim \text{Student-t}(n-q-1) \quad (\text{loi exacte}). \end{aligned}$$

Permet de calculer un IC avec couverture exacte pour  $n$  fini.

*Facteur d'inflation* de la variance  $(n-2)/(n-q-2) > 1$  dû à l'estimation de  $\beta^*$ . Si on a déjà  $q$  VC, l'ajout d'une nouvelle VC n'est rentable que si la valeur de  $(1 - R_{cX}^2)$  est réduite d'une fraction  $\geq 1/(n-q-2)$ .

#### 7.4.4 Expériences pilotes pour estimer $\beta^*$ ?

Pour avoir un estimateur sans biais de  $\beta^*$ , on peut faire une expérience pilote de  $n_0$  observations, calculer un estimateur  $\hat{\beta}_0$  et l'utiliser pour les  $n - n_0$  observations restantes. On obtient :

$$\begin{aligned} \bar{X}_{cp,n} &= \frac{1}{n-n_0} \sum_{i=n_0+1}^n (X_i - \hat{\beta}_0^T (C_i - \nu)) \quad \text{et} \\ S_{cp,n}^2 &= \frac{1}{(n-n_0-1)} \sum_{i=n_0+1}^n (X_i - \hat{\beta}_0^T (C_i - \nu) - \bar{X}_{cp,n})^2. \end{aligned}$$

On a  $E[\bar{X}_{cp,n}] = \mu$  et  $E[S_{cp,n}^2/(n-n_0)] = \text{Var}[\bar{X}_{cp,n}]$ .

Mais sous l'hypothèse de normalité,

$$\frac{\text{Var}[\bar{X}_{cp,n}]}{\text{Var}[\bar{X}_{ce,n}]} = \frac{n(n-q-2)(n_0-2)}{(n-n_0)(n-2)(n_0-q-2)} > 1.$$

C'est donc *inefficace*.

## 7.5 Espérance conditionnelle (EC) ou Monte Carlo conditionnel (CMC)

Un exemple simple. Supposons que l'on veut estimer  $\mu = P[Y_1 + \dots + Y_t > x]$ .

Estimateur évident :  $X = I[Y_1 + \dots + Y_t > x]$ .

Estimateur CMC :  $X_{e,s} = P[Y_1 + \dots + Y_t > x \mid Y_1, \dots, Y_s]$ , pour  $s \leq t$ .

Si  $s = t$  : aucun changement.

Si  $s = t - 1$  et les  $Y_j$  sont indépendants :  $X_{e,t-1} = 1 - F_t[Y_1 + \dots + Y_{t-1}]$ .

Si  $s = 0$  :  $X_{e,0} = P[Y_1 + \dots + Y_t > x] = \mu$  (variance réduite à zéro).

En fait plus  $s$  est petit, plus la variance est réduite.

### 7.5.1 Cadre général

*Idée* : remplacer l'estimateur  $X$  par  $E[X \mid Z]$  où  $Z$  est une autre v.a., ou plus généralement par  $E[X \mid \mathcal{G}]$ , où  $\mathcal{G}$  est une *information partielle* sur  $X$ .

L'estimateur CMC s'écrit

$$X_e \stackrel{\text{def}}{=} E[X \mid \mathcal{G}]$$

et on a

$$E[X_e] = E[E[X \mid \mathcal{G}]] = E[X].$$

De plus,

$$\text{Var}[X] = E[ \underbrace{\text{Var}[X \mid \mathcal{G}]}_{\substack{\text{Var. résiduelle} \\ \text{pour } \mathcal{G} \text{ connu,} \\ \text{éliminé par CMC}}} ] + \underbrace{\text{Var}[E[X \mid \mathcal{G}]]}_{\substack{\text{Var. due à la} \\ \text{variation de } \mathcal{G}}}$$

et donc

$$\text{Var}[X_e] = \text{Var}[X] - E[\text{Var}[X \mid \mathcal{G}]] \leq \text{Var}[X].$$

Pour minimiser la variance, on doit maximiser  $E[\text{Var}[X \mid \mathcal{G}]]$ , i.e.,  $\mathcal{G}$  doit contenir le moins d'information possible.

*Mais* moins  $\mathcal{G}_1$  contient d'information, plus il est difficile de calculer  $X_e$ .

On doit donc faire un *compromis*.

**Exemple.** *La banque.* (Exemple de Bratley, Fox, et Schrage 1987).

Une banque a 3 caissiers, mais parfois certains sont absents. Chaque jour,  $P[t \text{ caissiers présents}] = q_t$ . On a  $q_3 = 0.80$ ,  $q_2 = 0.15$ ,  $q_1 = 0.05$ .

La banque ouvre à 10h et ferme à 15h. Les clients arrivent de 9h45 à 15h, selon un processus de Poisson.

*Taux d'arrivée* : 1 par min de 11h à 14h, et 0.5 par min autrement. Ceux arrivant avant 10h doivent attendre. Le service débute à 10h. À 15h, la porte ferme, mais les clients déjà entrés seront servis. Il y a une seule *file d'attente* (FIFO), avec déroboade ("balking").

Un client qui arrive renonce avec probabilité  $p_k$  s'il y a  $k$  clients en face de lui. On a

$$p_k = \begin{cases} 0 & \text{si } k \leq 5; \\ (k - 5)/5 & \text{si } 5 < k < 10; \\ 1 & \text{si } k \geq 10. \end{cases}$$

Les *durées de service* sont indépendantes, Erlang (2, 1).

On veut estimer le *nombre espéré de clients* servis par jour.

Supposons que l'on simule  $n$  jours.

$T_i$  = Nb. de *caissiers* au jour  $i$  ;

$X_i$  = Nb. *servis* durant le jour  $i$  ;

$B_i$  = Nb. qui *renoncent* durant le jour  $i$  ;

$C_i = X_i + B_i$  = Nb. d'*arrivées* durant le jour  $i$ .

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i.$$

Soient :  $A_{i1}, S_{i1}, B_{i1}, A_{i2}, S_{i2}, B_{i2}, \dots$  :

les *interarrivées*, *durées de service*, et *renonciations* (0 ou 1) au jour  $i$ .

$X_i, B_i$  et  $C_i$  sont fonctions de

$$(T_i, A_{i1}, S_{i1}, B_{i1}, A_{i2}, S_{i2}, B_{i2}, \dots).$$

Soit  $B$  le nombre d'*abandons* dans une journée. On peut estimer  $E[B]$  par  $B$ . Supposons maintenant que l'on rend les *abandons invisibles*, i.e., on ne calcule pas les quantités liées aux individus qui abandonnent.  $\mathcal{G}$  représente tout le reste de l'information. L'Estimateur CMC est  $B_e = E[B | \mathcal{G}]$ .

Soit  $\lambda(t)$  le taux d'arrivée au temps  $t$  ;  
 $N(t)$  le nombre de clients dans le système au temps  $t$  ;  
 $P(t) = p_{N(t)}$ .

Le processus d'arrivée conditionnel (invisible) des clients qui abandonnent est un processus de Poisson de taux  $\{\lambda(t)P(t), t \geq 0\}$ .

L'espérance conditionnelle du nombre d'abandons est donc

$$B_e = \int_{9.75}^{15} \lambda(t)P(t)dt = 30 \int_{9.75}^{11} P(t)dt + 60 \int_{11}^{14} P(t)dt + 30 \int_{14}^{15} P(t)dt.$$

Facile à calculer et on sait que  $\text{Var}[B_e] < \text{Var}[B]$ .

## 7.6 Stratification

*Idée générale* : partitionner la "population" en  $k$  strates, de manière à ce que la variance à l'intérieur des strates soit inférieure à la variance générale, puis estimer la quantité désirée séparément à l'intérieur de chaque strate.

- $p_s$  = fraction de la population dans la strate  $s$ .
- $N_s$  = nombre d'observations dans la strate  $s$ .
- $n = \sum_{s=1}^k N_s$ .
- $X_{s,i}$  =  $i$ -ième observation de la strate  $s$ .

L'estimateur de

$$\mu = E[X] = \sum_{s=1}^k p_s E[X_{s,i}] = \sum_{s=1}^k p_s \mu_s$$

est alors

$$\bar{X}_{s,n} = \sum_{s=1}^k p_s \left( \frac{1}{N_s} \sum_{i=1}^{N_s} X_{s,i} \right) = \sum_{s=1}^k p_s \hat{\mu}_s.$$

Cas où on peut décider des strates :  $N_s = n_s$ .

Soit  $\sigma_s^2 = \text{Var}[X | S = s]$ , la variance dans la strate  $s$ . On a

$$\text{Var}[\bar{X}_{s,n}] = \sum_{s=1}^k p_s^2 \text{Var}[\hat{\mu}_s] = \sum_{s=1}^k p_s^2 \sigma_s^2 / n_s,$$

dont un estimateur sans biais est :

$$S_{s,n}^2 = \sum_{s=1}^k p_s^2 \hat{\sigma}_s^2 / n_s,$$

où  $\hat{\sigma}_s^2$  est la variance empirique  $X_{s,1}, \dots, X_{s,n_s}$ .

*Allocation proportionnelle* :  $n_s \approx np_s$ .

$$\text{Var}[\bar{X}_{\text{sp},n}] = \frac{1}{n} \sum_{s=1}^k p_s \sigma_s^2.$$

*Allocation optimale* : On veut minimiser  $\text{Var}[\bar{X}_{s,n}]$  p.r. à  $n_1, \dots, n_k$ , sous les contraintes que  $n_s \geq 0$  et  $n_1 + \dots + n_k = n$ .

En utilisant un multiplicateur de Lagrange, on obtient (en négligeant les arrondis) :

$$n_s^* = np_s \sigma_s / \bar{\sigma}$$

où

$$\bar{\sigma} = \sum_{s=1}^k p_s \sigma_s.$$

Cela donne

$$\text{Var}[\bar{X}_{\text{so},n}] = \sum_{s=1}^k \frac{p_s^2 \sigma_s^2}{n_s} = \sum_{s=1}^k \frac{p_s^2 \sigma_s^2}{np_s \sigma_s / \bar{\sigma}} = \frac{\bar{\sigma}^2}{n}.$$

En décomposant la variance, on peut comparer :

$$\begin{aligned} \text{Var}[\bar{X}_n] &= \frac{1}{n} (E[\text{Var}[X|S]] + \text{Var}[E[X|S]]) \\ &= \frac{1}{n} \left( \sum_{s=1}^k p_s \sigma_s^2 + \sum_{s=1}^k p_s (\mu_s - \mu)^2 \right) \\ &= \text{Var}[\bar{X}_{\text{sp},n}] + \frac{1}{n} \sum_{s=1}^k p_s (\mu_s - \mu)^2 \\ &= \text{Var}[\bar{X}_{\text{so},n}] + \frac{1}{n} \sum_{s=1}^k p_s (\sigma_s - \bar{\sigma})^2 + \frac{1}{n} \sum_{s=1}^k p_s (\mu_s - \mu)^2. \end{aligned}$$

*Post-stratification* : Ne pas choisir (fixer) les  $N_s$  ; laisser la simulation les générer comme d'habitude, puis utiliser  $\bar{X}_{s,n}$ .

Avantages : Dans plusieurs situations, on ne peut pas (ou il est trop difficile) de choisir la strate. On peut appliquer la post-stratification seulement après coup.

## 7.7 Variables antithétiques (AV)

Semblable à l'idée des VAC, sauf que l'on veut maintenant estimer *une seule espérance* par un moyenne de plusieurs estimateurs *négativement corrélés*.

Supposons que l'on a  $k$  estimateurs sans biais de  $\mu$ ,  $(X^{(1)}, \dots, X^{(k)})$ , avec  $\text{Var}[X^{(1)}] = \dots = \text{Var}[X^{(k)}]$ . La moyenne

$$X_a = \frac{1}{k} \sum_{\ell=1}^k X^{(\ell)}$$

est un estimateur *sans biais* de  $\mu$ , avec variance

$$\text{Var}[X_a] = \frac{1}{k^2} \sum_{j=1}^k \sum_{\ell=1}^k \text{Cov}[X^{(j)}, X^{(\ell)}] = \frac{\text{Var}[X^{(1)}]}{k} + \frac{2}{k^2} \sum_{j<\ell} \text{Cov}[X^{(j)}, X^{(\ell)}].$$

Si les  $X^{(\ell)}$  sont indépendants, les covariances sont nulles. La variance est réduite ssi  $\text{Var}[X_a] < \text{Var}[X^{(1)}]/k$ , ssi la somme des covariances est négative.

Pour un total de  $n = mk$  répétitions, on définit

$$\bar{X}_{a,n} = \frac{1}{m} \sum_{i=1}^m X_{a,i} = \frac{1}{n} \sum_{i=1}^m \sum_{\ell=1}^k X_i^{(\ell)},$$

où  $X_{a,1}, \dots, X_{a,m}$  sont des copies i.i.d. de  $X_a$ .

Estimateur sans biais de  $\text{Var}[X_a]$  : variance empirique des  $m$  copies,

$$S_{a,m}^2 = \frac{1}{m-1} \sum_{i=1}^m (X_{a,i} - \bar{X}_{a,n})^2.$$

### 7.7.1 Méthode AV générale

Soit  $\mu = E[X]$ , où  $X = f(\mathbf{U})$  et  $\mathbf{U} = (U_1, U_2, \dots)$  une suite i.i.d.  $U(0, 1)$ . Pour  $\ell = 1, \dots, k$ , soit  $X^{(\ell)} = f(\mathbf{U}^{(\ell)})$  où  $\mathbf{U}^{(\ell)}$  est une suite i.i.d.  $U(0, 1)$ . Le but de l'approche de variables antithétiques généralisées est d'induire une dépendance entre les  $\mathbf{U}_\ell$  de manière à rendre  $\sum_{j < \ell} \text{Cov}[X^{(j)}, X^{(\ell)}] < 0$ .

### 7.7.2 Méthode AV "classique" : Paires antithétiques.

On a  $k = 2$ ,  
 $\mathbf{U}^{(1)} = \mathbf{U} = (U_1, U_2, \dots)$ ,  
 $\mathbf{U}^{(2)} = \mathbf{1} - \mathbf{U} = (1 - U_1, 1 - U_2, \dots)$ , la suite antithétique.

Posons  $X_a = [X^{(1)} + X^{(2)}]/2$ ,

$$\begin{aligned} \text{Var}[X_a] &= (\text{Var}[X^{(1)}] + \text{Var}[X^{(2)}] + 2 \text{Cov}[X^{(1)}, X^{(2)}])/4 \\ &= (\text{Var}[X^{(1)}] + \text{Cov}[X^{(1)}, X^{(2)}])/2, \end{aligned}$$

et  $\text{Var}[X_a] < \text{Var}[X]/2$  ssi  $\text{Cov}[X^{(1)}, X^{(2)}] < 0$ .

*Intuition* : Les événements désastreux pour  $X^{(1)}$  seront compensés par des événements antithétiques chanceux pour  $X^{(2)}$ , et vice-versa.

### 7.7.3 Combinaison VAC-AV

Supposons que l'on veut comparer deux systèmes en utilisant des VAC, et aussi des paires de répétitions antithétiques.

Si ces deux méthodes fonctionnent bien séparément, fera-t-on nécessairement mieux en les combinant ? *Non*.

Soit  $X_k^{(1)} = f_k(\mathbf{U})$  et  $X_k^{(2)} = f_k(\mathbf{1} - \mathbf{U})$  la paire antithétique pour le système

k. On a

$$\begin{aligned}
& \text{Var} \left[ \frac{X_2^{(1)} + X_2^{(2)}}{2} - \frac{X_1^{(1)} + X_1^{(2)}}{2} \right] \\
&= \frac{\text{Var}[X_1^{(1)}] + \text{Var}[X_1^{(2)}] + \text{Var}[X_2^{(1)}] + \text{Var}[X_2^{(2)}]}{4} \\
&\quad + \frac{\text{Cov}[X_1^{(1)}, X_1^{(2)}] + \text{Cov}[X_2^{(1)}, X_2^{(2)}]}{2} - \frac{\text{Cov}[X_1^{(1)}, X_2^{(1)}] + \text{Cov}[X_1^{(2)}, X_2^{(2)}]}{2} \\
&\quad - \frac{\text{Cov}[X_1^{(2)}, X_2^{(1)}] + \text{Cov}[X_1^{(1)}, X_2^{(2)}]}{2}. \\
&= \text{Var}[X_1^{(1)}] + \underbrace{\text{Cov}[X_1^{(1)}, X_1^{(2)}]}_{\leq 0} - \underbrace{\text{Cov}[X_1^{(1)}, X_2^{(1)}]}_{\geq 0} - \underbrace{\text{Cov}[X_1^{(2)}, X_2^{(1)}]}_{?}.
\end{aligned}$$

## 7.8 Méthodes QMCR

Bien que les méthodes quasi-Monte Carlo randomisées (QMCR) soient des techniques de variables aléatoires antithétiques généralisées, elles méritent une section séparée en raison de leur importance. L'idée de base est de partir d'un ensemble  $P_n = \{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\}$  de  $n$  points qui couvrent l'hypercube unité  $[0, 1]^t$  de manière très uniforme, et de perturber aléatoirement  $P_n$  de sorte qu'après la perturbation

(R1) il conserve sa forte uniformité quand pris comme ensemble, et

(R2) chaque point individuel possède la distribution uniforme sur  $[0, 1]^t$ .

Dénotons l'ensemble de points randomisés par  $\tilde{P}_n = \{\mathbf{U}_0, \dots, \mathbf{U}_{n-1}\}$ . L'estimateur de  $\mu = E[f(\mathbf{U})]$  basé sur une copie de la randomisation est

$$X_{\text{rqmc}} = Q_n = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{U}_i). \quad (7.1)$$

Cette randomisation est répétée  $m$  fois, indépendamment, pour un certain entier positif  $m$ , avec le même  $P_n$ . Supposons que n'importe quels points pris de différentes randomisations sont indépendants deux à deux, dans le sens que si  $\mathbf{U}_i^{(j)}$  est le  $i^e$  point de la  $j^e$  randomisation, alors

(R3) pour tout  $i_1, i_2$  et  $j_1 \neq j_2$ ,  $\mathbf{U}_{i_1}^{(j_1)}$  et  $\mathbf{U}_{i_2}^{(j_2)}$  sont indépendants.

Dénotons par  $X_{\text{rqmc},1}, \dots, X_{\text{rqmc},m}$  les  $m$  copies de  $X_{\text{rqmc}}$  obtenues par ces randomisations, et soit  $\bar{X}_{\text{rqmc},m}$  et  $S_{\text{rqmc},m}^2$  leur moyenne et variance échantillonnales. Nous avons alors la proposition suivante.

**Proposition 5.** *Sous les conditions (R1) à (R3),  $E[\bar{X}_{\text{rqmc},mn}] = \mu$  et*

$$E[S_{\text{rqmc},m}^2] = \text{Var}[X_{\text{rqmc}}] = m \text{Var}[\bar{X}_{\text{rqmc},mn}].$$

Par conséquent, nous obtenons des variables uniformes corrélées, du fait de la couverture uniforme obtenue avec  $P_n$ , comme souhaité dans les techniques de variables antithétiques généralisées. Cette couverture uniforme doit être conservée autant que faire se peut lors de la transformation d'un vecteur d'uniformes en vecteur aléatoire. Les techniques basées sur l'inversion parviennent généralement mieux à respecter cette exigence par rapport aux autres méthodes abordées au Chapitre 5. Deux questions se posent à présent : comment construire l'ensemble initial  $P_n$ , et comment effectuer adéquatement la perturbation aléatoire.

### 7.8.1 Techniques quasi-Monte Carlo

En contraste avec la méthode Monte Carlo, les approches quasi-Monte Carlo classiques se basent sur des ensembles déterministes de points, soigneusement construits. Ces ensembles de points peuvent être les  $n$  premiers points d'une séquence infinie, où les points sont énumérés dans un ordre spécifique. La construction des points  $x_i$  dépend aussi explicitement de la dimension du problème ; en particulier, les vecteurs  $x_i$  dans  $[0, 1]^s$  ne peuvent pas être construits en prenant des ensembles de  $d$  éléments consécutifs d'une séquence scalaire, contrairement au Monte Carlo. De plus, sans une borne supérieure sur  $s$ , les méthodes quasi-Monte Carlo sont inapplicables.

Nous énumérerons dans cette section les points d'un ensemble de points  $P_n \in [0, 1]^s$  de 0 à  $n - 1$ , afin d'être consistant avec la notation habituelle dans la littérature quasi-Monte Carlo, qui comprend de nombreuses références. Nous nous inspirons ici partiellement du Chapitre 8 de Glasserman [7], et renvoyons le lecteur à [9].

#### Règle de treillis de rang 1

Une règle de treillis (lattice) de rang 1 avec  $n$  points en  $d$  dimensions est définie comme suit. Choisissons un vecteur  $\mathbf{a}_1 = (a_1, \dots, a_d)$  dont les coordonnées appartiennent à  $\mathcal{Z}_n = \{0, \dots, n-1\}$ , et posons  $\mathbf{v}_1 = \mathbf{a}_1/n$ . Nous définissons l'ensemble de points  $P_n = \{\mathbf{v} = i\mathbf{v}_1 \bmod 1, i = 0, 1, \dots, n-1\}$ , où la division et l'opération modulo se font coordonnée par coordonnée. Cet ensemble de points est l'intersection d'un treillis avec l'hypercube unité en  $d$  dimensions. Les  $a_j$  sont généralement

pris premiers relativement à  $n$ , de sorte que la projection de  $P_n$  sur n'importe quelle coordonnée contient  $n$  points distincts, à savoir  $\{0, 1/n, \dots, (n-1)/n\}$ . Un cas particulier sont les règles de Korobov, qui sont des règles de treillis de rang-1 avec  $\mathbf{a}_1 = (1, a, a^2, \dots, a^{d-1})$ .

Le vecteur  $\mathbf{v}_1$  est sélectionné afin de minimiser une mesure donnée de discrédance de  $P_n$ . Cette discrédance peut par exemple être le critère  $P_\alpha$  pondéré, avec des poids produits, dont le carré est

$$D^2(P_n) = -1 + \frac{1}{n} \sum_{i=0}^{n-1} \prod_{j=1}^d \left( 1 + \gamma_j \frac{(-4\pi^2)^\alpha B_{2\alpha}(u_j)}{(2\alpha)! 2} \right),$$

où  $\alpha$  et les  $\gamma_j$  sont des nombres réels positifs, et  $B_{2\alpha}$  est le polynôme de Bernoulli polynomial d'ordre  $2\alpha$ .

### Réseau de Sobol

Un réseau de Sobol avec  $n = 2^k$  points en  $d$  dimensions, pour un certain entier positif  $k$ , contient les premiers  $n$  points d'une séquence de Sobol en  $d$  dimensions. Ces points sont définis par des matrices binaires  $w \times k$   $\mathbf{C}_1, \dots, \mathbf{C}_d$ , appelées matrices génératrices, où  $w$  est le nombre de bits retenus dans le développement binaire des points. Afin de définir le  $i^e$  point  $\mathbf{u}_i$ , pour  $i = 0, \dots, 2^k - 1$ , nous écrivons le développement digital de  $i$  en base 2 et multiplions le vecteurs de ses composantes par  $\mathbf{C}_j$ , modulo 2, afin d'obtenir les composantes du développement binaire de la  $j^e$  coordonnée  $\mathbf{u}_i$ . Les matrices  $\mathbf{C}_j$  ne sont pas uniques; elles dépendent du choix des nombres de direction, qui déterminent leurs colonnes.

#### 7.8.2 Perturbations aléatoires

La manière la plus simple de randomiser  $P_n$  est la translation aléatoire modulo 1, qui consiste dans la génération d'un seul vecteur  $\mathbf{U}$  uniformément sur  $(0, 1)^d$ , et à l'ajout de celui-ci à chaque point de  $P_n$ , modulo 1, coordonnée par coordonnée. Dans le cas de treillis de rang-1, nous pouvons par la suite effectuer une transformation du boulanger, qui remplace chaque coordonnée  $u$  de chaque point par  $2u$  si  $u < 1/2$  et par  $2 - 2u$  sinon. Cette transformation périodise l'intégrand et intègre exactement les composantes linéaires.

Nous pouvons randomiser les réseaux de Sobol au moyen d'un scramble matriciel à gauche, suivi d'une translation aléatoire digitale. Le scramble matriciel

à gauche multiplie chaque matrice  $C_j$  (modulo 2) par une matrice binaire triangulaire inférieure  $M_j$  avec des 1's sur la diagonale et des bits aléatoires sous la diagonale. La translation aléatoire digitale génère un seul point  $U$  uniformément sur  $(0, 1)^d$  et ajoute chaque digits du développement binaire de chacune de ses coordonnées au digit correspondant de chaque point de  $P_n$ , modulo 2.

## 7.9 Exercices

1. Considérons l'intégration

$$\int_0^1 \frac{1}{1+x} dx,$$

qui donne bien entendu  $\ln 2$ . Il est toutefois possible d'estimer cette intégrale par simulation, comme elle correspond à

$$E \left[ \frac{1}{1+X} \right],$$

avec  $X \sim U(0, 1)$ . Estimez cette intégrale avec  $n = 1500$  tirs, et analysez l'apport de la variable de contrôle  $Y = 1 + X$ . Que donne l'utilisation de variables antithétiques pour ce problème ?

## Annexe A

# Rappels de Statistiques et de Probabilités

Nous récapitulons ici brièvement quelques concepts importants en statistiques et en probabilités, ainsi que diverses distributions usuelles. Il est en effet nécessaire de disposer d'un minimum de connaissances dans ce domaine lorsque nous nous intéressons à la simulation stochastique. En particulier, les probabilités et les statistiques sont des outils de base pour comprendre comment modéliser un système probabiliste, valider le modèle de simulation, choisir les distributions de probabilités, générer des échantillons aléatoires à partir de ces distributions, réaliser des analyses statistiques sur les sorties de la simulation et concevoir les expériences de simulation. La lecture des Chapitres 4 et 6 du livre de Law [8] représentera un complément appréciable.

### A.1 Variables aléatoires

Une expérience aléatoire est une action ou une observation dont le résultat est incertain préalablement à sa réalisation. Un espace de probabilité est un triplet  $(\Omega, \mathcal{F}, P)$ , où  $\Omega$ , l'espace d'échantillonnage, est l'ensemble de toutes les réalisations possibles de l'expérience, appelés éléments aléatoires, individuellement dénotés par  $\omega$ . La collection  $\mathcal{F}$  des événements aléatoires est un  $\sigma$ -champ des sous-ensembles de  $\Omega$ . Nous dirons que l'événement  $A \in \mathcal{F}$  est réalisé si le résultat de l'expérience est un élément de  $A$ . Une mesure  $P$  est assignée aux éléments de  $\mathcal{F}$ ,  $P(A)$  étant la probabilité de  $A$ . Formellement, nous avons la définition ci-après.

**Definition A.1:** Mesure de probabilité

Une *mesure de probabilité* sur un espace mesurable  $(\Omega, \mathcal{F})$  est une fonction d'ensemble  $P : \mathcal{F} \rightarrow [0, 1]$  satisfaisant les axiomes de probabilité :

- (a)  $P[A] \geq 0$ , pour tout  $A \in \mathcal{F}$  ;
- (b)  $P[\Omega] = 1$  ;
- (c) additivité dénombrable : pour une collection disjointe  $\{A_j \in \mathcal{F}, j \in \mathbb{N}\}$ ,

$$P[\cup_j A_j] = \sum_j P[A_j].$$

## A.2 Quelques caractérisations des distributions.

Une variable aléatoire (v.a.) est une fonction qui assigne un nombre réel à chaque point d'un espace d'échantillonnage  $\Omega$ . Plus formellement, une v.a. est une fonction  $\mathcal{F}/\mathcal{B}$ -mesurable  $X : \Omega \rightarrow \mathbb{R}$ , où  $\mathcal{B}$  est le champ linéaire de Borel. Nous utiliserons les abréviations et notations suivantes :

**i.i.d.** indépendamment et identiquement distribuées.

**IC** intervalle de confiance.

### A.2.1 Variables discrètes et continues

Soit  $X$  une v.a. univariée.  $X$  est dite discrète si elle peut prendre au plus un nombre dénombrable de valeurs  $x_1, x_2, \dots$ . La probabilité que la variable aléatoire  $X$  prenne la valeur  $x_i$  est donnée par

$$p(x_i) = P[X = x_i], \text{ pour } i = 1, 2, \dots$$

$p(x)$  est appelée fonction de masse de probabilité, et de la définition A.1, nous avons

$$\sum_{i=1}^{\infty} p(x_i) = 1.$$

Une variable aléatoire  $X$  est dite continue s'il existe une fonction nonnégative

$f(x)$  telle que pour n'importe quel ensemble de nombres réels  $B$ ,

$$P[X \in B] = \int_B f(x)dx, \text{ et } \int_{-\infty}^{\infty} f(x)dx = 1.$$

$f(x)$  est appelée fonction de densité de probabilité de la v.a. continue  $X$ .

### A.2.2 Fonction de répartition

La fonction de répartition est définie comme suit

$$F_X(x) = P[X \leq x] = \begin{cases} \sum_{j=1}^K p(x_j) & \text{(cas discret),} \\ \int_{-\infty}^x f(y)dy & \text{(cas continu),} \end{cases}$$

où  $K = \max k$  t.q.  $x_k \leq x$ , et la fonction de survie :

$$\bar{F}_X(x) = P[X > x].$$

Nous utiliserons généralement  $F(x)$  au lieu de  $F_X(x)$ , quand le contexte permet d'éviter toute confusion quant à la variable aléatoire  $X$  concernée. La fonction de répartition est par conséquent monotone croissante, et continue à droite. De plus, nous avons

$$\lim_{x \rightarrow -\infty} F(x) = 0, \text{ et } \lim_{x \rightarrow \infty} F(x) = 1.$$

La fonction de répartition  $F(x)$  pour une variable aléatoire continue  $X$  est aussi donnée par

$$F(x) = P[X \in (-\infty, x]] = \int_{-\infty}^x f(y)dy, \forall -\infty < x < \infty.$$

Par conséquent (moyenne certaines conditions de régularités légères),  $f(x) = F'(x)$ . De plus, si  $I = [a, b]$  pour n'importe quels nombres réels  $a$  et  $b$  tels que  $a < b$ ,

$$P[X \in I] = \int_a^b f(y)dy = F(b) - F(a).$$

Le paquetage `probdist` de `SSJ` offre un large support des distributions usuelles, discrètes ou continue.

### A.2.3 Propriétés

Espérance  $\mu = E[X]$ ;

$$E[X] = \begin{cases} \sum_{i=1}^{\infty} x_i p(x_i) & \text{dans le cas discret;} \\ \int_{-\infty}^{\infty} x f(x) dx & \text{dans le cas continu.} \end{cases}$$

Variance  $\sigma^2 = E[(X - \mu)^2]$ ;

Écart-type  $\sigma$ ;

Coefficient de variation  $\sigma/\mu$ ;

Coefficient d'asymétrie  $\nu = E[(X - \mu)^3]/\sigma^3$

Coefficient d'aplatissement  $\kappa = E[(X - \mu)^4]/\sigma^4$ .

Si  $\nu = 0$ , la loi symétrique ; si  $\nu > 0$ , elle est asymétrique à droite.

Pour certaines familles de loi (e.g., Johnson), on peut choisir les paramètres pour obtenir des valeurs arbitraires de ces 4 coefficients.

### A.2.4 Convergence de variables aléatoires

En théorie des probabilités, il existe différentes notions de convergence de variables aléatoires. Dans la suite, nous supposons que  $(X_n)$  est une suite de variables aléatoires réelles, et  $X$  est une variable aléatoire réelle.

**Convergence en loi (ou en distribution)** Soient  $F_1, F_2, \dots$  la suite de fonctions de répartition associées aux variables aléatoires réelles  $X_1, X_2, \dots$ , et  $F$  la fonction de répartition associée à la variable aléatoire réelle  $X$ . La suite  $(X_n)$  converge vers  $X$  en loi, si

$$\lim_{n \rightarrow \infty} F_n(a) = F(a),$$

pour tout réel  $a$  auquel  $F$  est continue. La convergence en loi est souvent notée en ajoutant la lettre  $\mathcal{L}$  (ou  $\mathcal{D}$  pour distribution) au-dessus de la flèche de convergence :

$$X_n \xrightarrow{\mathcal{L}} X.$$

**Convergence en probabilité**  $X_n$  converge vers  $X$  en probabilité ssi

$$\lim_{n \rightarrow \infty} P(|X_n - X| \geq \varepsilon) = 0, \text{ pour tout } \varepsilon > 0.$$

La convergence en probabilité implique la convergence en loi.

**Convergence presque sûre (ou avec probabilité 1)** On dit que  $X_n$  converge presque sûrement ou presque partout ou avec probabilité 1 ou fortement vers  $X$  si

$$P\left[\lim_{n \rightarrow \infty} X_n = X\right] = 1.$$

La convergence presque sûre implique la convergence en probabilité, et donc implique la convergence en loi. Il s'agit par conséquent de la la notion la plus forte de convergence, et nous utiliserons régulièrement l'abréviation a.p. 1 pour désigner le terme "avec probabilité 1".

### A.3 Quelques lois de probabilité discrètes

#### A.3.1 Loi uniforme discrète

Paramètres  $(i, j)$

$$P[X = x] = 1/(j - i + 1) \text{ pour } x = i, i + 1, \dots, j.$$

#### A.3.2 Loi binomiale

Paramètres  $(n, p)$

$$P[X = x] = \binom{n}{x} p^x (1 - p)^{n-x} \text{ pour } x = 0, \dots, n,$$

où

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

On a  $E[X] = np$  et  $\text{Var}[X] = np(1 - p)$ . Dans le cas particulier  $n = 1$ , nous obtenons la loi de Bernoulli.

#### A.3.3 Loi multinomiale

La loi multinomiale est la généralisation directe de la loi binomiale, en considérant  $m$  résultats possibles au lieu de 2. Nous considérons alors les variables aléatoires  $N_i, i = 1, \dots, m$ , le nombre d'occurrences du  $i^e$  événement possible. Chaque

Domaine	$0 \leq x < \infty.$
Paramètres	$n > 0, n \in \mathbb{N},$ $p_1, \dots, p_m,$ tels que $\sum_{i=1}^m p_i = 1.$
Masse de probabilité	$\frac{n!}{n_1! \dots n_m!} p_1^{n_1} \dots p_m^{n_m}$

TABLE A.1 – Distribution multinomiale

événement est associé à la probabilité  $p_i, i = \{1, \dots, m\}$ , avec les contraintes  $\sum_{i=1}^m N_i = n$  et  $\sum_{i=1}^m p_i = 1.$

Lorsque la variable aléatoire  $N_i$  tend vers l'infini, en vertu du théorème de la limite, nous obtenons

$$\frac{N_i - np_i}{\sqrt{np_i(1 - p_i)}} \Rightarrow N(0, 1),$$

de sorte que

$$\sum_{i=1}^m \frac{(N_i - np_i)^2}{np_i} \Rightarrow \chi_{(m-1)}^2.$$

Cette dernière remarque est à la base du test du  $\chi^2.$

### A.3.4 Loi géométrique

La distribution géométrique prend un paramètre  $p,$  et a comme masse de probabilité<sup>1</sup>

$$P[X = x] = p(1 - p)^x \text{ pour } x = 0, 1, \dots$$

Elle jouit de la propriété sans sans mémoire, à savoir  $P[X = y + x | X \geq y] = P[X = x].$

La fonction de répartition est

$$F(x) = \sum_{y=0}^{[x]} P[X = y] = 1 - (1 - p)^{1+[x]} \quad \text{pour } x \geq 0.$$

et la moyenne et la variance sont données par  $E[X] = (1 - p)/p, \text{ Var}[X] = (1 - p)/p^2.$

La loi géométrique peut s'interpréter comme le nombre d'expériences répétées jusqu'à on obtenir un premier succès. A chaque essai, la probabilité d'un tel succès

1. certains auteurs préfèrent considérer  $x \geq 1,$  auquel cas nous aurons  $P[X = x] = p(1 - p)^{x-1}$  pour  $x = 1, \dots$

est  $p$ , autrement dit la loi géométrique permet de caractériser un processus de Bernoulli qui est arrêté dès un succès rencontré.

### A.3.5 Loi de Poisson

Paramètre ( $\lambda$ )

$$P[X = x] = \frac{\lambda^x e^{-\lambda}}{x!} \text{ pour } x = 0, 1, 2, \dots$$

$$E[X] = \text{Var}[X] = \lambda.$$

Si  $X_1, \dots, X_q$  indep.,  $X_i \sim \text{Poisson}(\lambda_i)$ , alors  $X = X_1 + \dots + X_q \sim \text{Poisson}(\lambda)$  où  $\lambda = \lambda_1 + \dots + \lambda_q$ .

Si  $X \sim \text{Binomiale}(n, p)$  et  $(n, np) \rightarrow (\infty, \lambda)$ , alors  $X \Rightarrow \text{Poisson}(\lambda)$ .  
Adapté pour représenter les événements rares.

**Theorem 14.** (Bornes explicites.) Soient  $X_1, \dots, X_n$  indep.,  $X_j \sim \text{Bernoulli}(p_j)$ ,  $X = X_1 + \dots + X_n$  et  $\lambda = p_1 + \dots + p_n$ .

Alors, pour tout  $x$ ,

$$\left| P[X = x] - \frac{\lambda^x e^{-\lambda}}{x!} \right| \leq \sum_{j=1}^n p_j^2 [= \lambda/n \text{ si } p_j = p \text{ pour tout } j].$$

Exemples : nombre d'erreurs typographiques dans un livre, nombre de personnes achetant des actions de Bombardier un jour donné, etc.

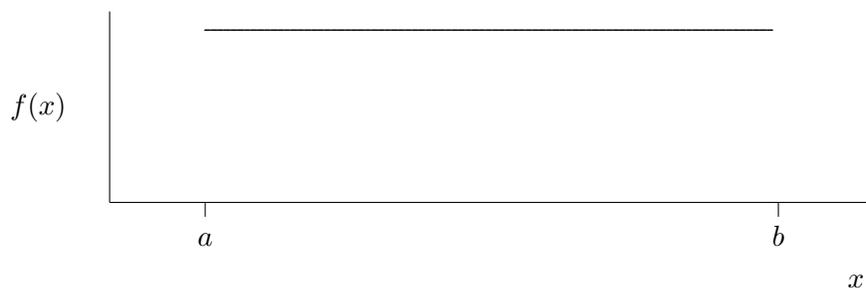
## A.4 Lois continues univariées

### A.4.1 Loi uniforme

Uniforme sur  $(a, b)$

$$f(x) = 1/(b - a) \text{ pour } a < x < b.$$

$$E[X] = (b - a)/2 \text{ et } \text{Var}[X] = (b - a)^2/12.$$



### A.4.2 Loi triangulaire

La distribution triangulaire est unimodale, avec une fonction de densité de probabilité qui est linéaire par morceaux. La fonction de densité est croissante sur un intervalle  $[a, c]$ , décroissante sur un intervalle  $[c, b]$ , et nulle ailleurs. Elle est souvent utilisée en l'absence de données.

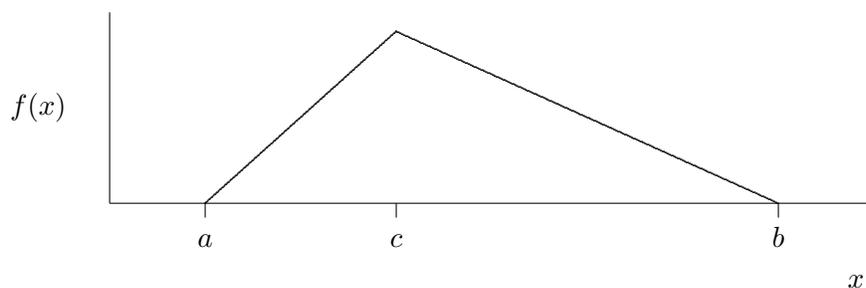


FIGURE A.1 – Fonction de densité de la distribution triangulaire

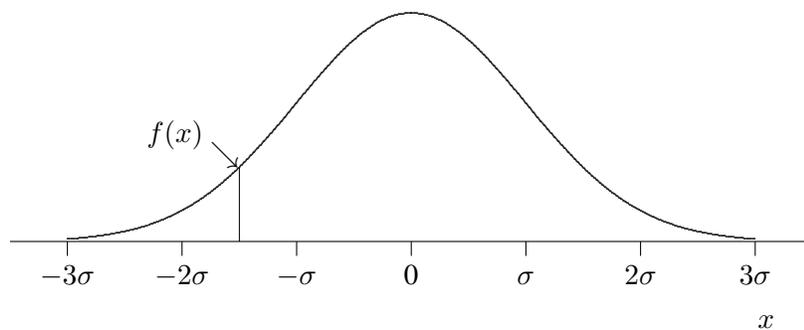
### A.4.3 Loi normale

Paramètres :  $(\mu, \sigma^2)$ ;  $X \sim N(\mu, \sigma^2)$ , de densité

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)} \text{ pour } x \in \mathcal{R}.$$

Domaine	$a \leq x \leq b.$
Paramètres	$a$ : limite inférieure. $b > a$ : limite supérieure. $c, a \leq c \leq b$ : mode.
Densité	$f(x) = \begin{cases} \frac{2}{b-a} \frac{x-a}{c-a} & \text{if } a \leq x \leq c, \\ \frac{2}{b-a} \frac{b-x}{b-c} & \text{if } c \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases}$
Fonction de répartition	$F(x) = \begin{cases} \frac{(x-a)^2}{(b-a)(c-a)} & \text{if } a \leq x \leq c, \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & \text{if } c \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases}$
Moyenne	$\frac{a+b+c}{3}$
Variance	$\frac{a^2+b^2+c^2-ab-ac-bc}{18}$
Mode	$c$
Médiane	$\begin{cases} a + \sqrt{\frac{(b-a)(c-a)}{2}} & \text{if } c \geq \frac{b-a}{2} \\ b - \sqrt{\frac{(b-a)(b-c)}{2}} & \text{if } c \leq \frac{b-a}{2} \end{cases}$

TABLE A.2 – Distribution triangulaire



Une propriété intéressante est la linéarité de la distribution normale : si  $X \sim N(\mu, \sigma^2)$ , alors  $Y = aX + b \sim N(a\mu + b, a^2\sigma^2)$ .

$$\Phi(x) = P[Z \leq x] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-s^2/2} ds.$$

Nous pouvons écrire en particulier.

$$X = \mu + \sigma N(0, 1).$$

Une des principales motivations derrière la loi normale se trouve dans le théorème de la limite centrale (TLC), qui énonce que  $X$  suit approximativement une distribution normale si  $X$  est une somme de nombreux petits effets indépendants.

#### A.4.4 Loi lognormale

Paramètres :  $(\mu, \sigma^2)$

Au lieu d'une somme, on a un *produit* de petits effets :  $Y_n = (Z_1 \cdot Z_2 \cdots Z_n)^{1/n}$ .  
On a  $\ln Y_n = (\ln Z_1 + \cdots + \ln Z_n)/n$ .

Sous les conditions du TLC :

$$\frac{\ln Y_n - E[\ln Y_n]}{(\text{Var}[\ln Y_n])^{1/2}} \Rightarrow N(0, 1) \text{ pour } n \rightarrow \infty.$$

On dit que  $X \sim \text{lognormale}$  si  $\ln X \sim \text{normale}$ .  
Cette loi apparaît souvent en économie et finance.

Si  $\ln X \sim N(\mu, \sigma^2)$ ,  $X \sim \text{lognormale}(\mu, \sigma^2)$  a la densité

$$f(x) = \frac{1}{\sqrt{2\pi\sigma x}} e^{-(\ln x - \mu)^2 / (2\sigma^2)} \text{ pour } x > 0.$$

On a  $E[X] = e^{\mu + \sigma^2/2}$  et  $\text{Var}[X] = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$ .  
Attention :  $E[\ln X] = \mu < \mu + \sigma^2/2 = \ln E[X]$ .

Attention à ne pas abuser de la lognormale. On l'utilise souvent pour des grandeurs supposées strictement positives.

Exemple : valeur du temps. Elle est souvent représentée par l'opposée d'une lognormale, mais cela cause beaucoup de problèmes d'estimation.

#### A.4.5 Loi de Erlang

Une variable aléatoire de Erlang est une variable gamma où le paramètre de forme  $\alpha$  est un entier. Elle correspond aussi à la somme de  $\alpha$  variables exponentielles i.i.d., de paramètre  $\lambda$ .

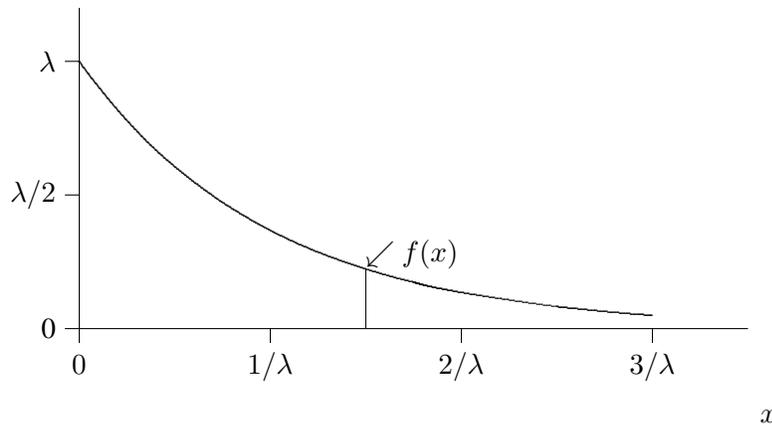
Domaine	$0 \leq x < \infty$ .
Paramètres	$\alpha \in \mathbb{N}_0$ : paramètre de forme ; $\lambda > 0$ : paramètre d'échelle.
Distribution	$1 - e^{-\lambda x} \left( \sum_{i=0}^{\alpha-1} \frac{(\lambda x)^i}{i!} \right)$
Densité	$\frac{\lambda(\lambda x)^{\alpha-1} e^{-\lambda x}}{(\alpha-1)!}$
Moyenne	$\frac{\alpha}{\lambda}$
Variance	$\frac{\alpha}{\lambda^2}$

TABLE A.3 – Distribution de Erlang

Domaine	$0 < x < \infty$ .
Paramètre	$\lambda > 0$ : scale parameter.
Distribution	$1 - e^{-\lambda x}$
Densité	$\lambda e^{-\lambda x}$
Moyenne	$\frac{1}{\lambda}$
Variance	$\frac{1}{\lambda^2}$

TABLE A.4 – Distribution exponentielle

#### A.4.6 Loi exponentielle



Il s'agit de la seule loi continue qui jouit de la propriété sans mémoire :

$$P[X > t + x \mid X > t] = \frac{P[X > t + x]}{P[X > t]} = \frac{e^{-\lambda(t+x)}}{e^{-\lambda t}} = e^{-\lambda x} = P[X > x].$$

Cette propriété simplifie beaucoup l'analyse mathématique et l'explique la grande popularité de la loi exponentielle.

Domaine	$-\infty < x < \infty.$
Paramètres	$a$ : paramètre de position, la médiane. $b$ : paramètre d'échelle.
Distribution	$\frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x-a}{b}\right)$
Densité	$\frac{1}{\pi b \left(1 + \left(\frac{x-a}{b}\right)^2\right)}$
Moments	n'existent pas.
Médiane	$a$
Mode	$a$

TABLE A.5 – Distribution de Cauchy

Exemple : durée de vie. Taux de panne constant.

Si  $X_1, \dots, X_k$  indep.,  $X_j \sim \text{exponentielle}(\lambda_j)$ ,  
alors  $X = \min(X_1, \dots, X_k) \sim \text{exponentielle}(\lambda)$  où  $\lambda = \lambda_1 + \dots + \lambda_k$ .

#### A.4.7 Loi de Cauchy

La distribution de Cauchy est unimodale et symétrique, avec des queues nettement plus épaisses que la normale. La fonction de densité de probabilité est symétrique autour de  $a$ , avec pour quartiles supérieur et inférieur  $a \pm b$ . Le rapport de deux variables aléatoires normales, indépendantes, et de moyennes nulles, suit une distribution de Cauchy.

La loi de Cauchy est de plus une des lois pour lesquelles le loi des grands nombres ne s'applique pas. Si  $X_1, \dots, X_n$  sont i.i.d. et suivent une distribution de Cauchy, la moyenne empirique

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$$

ne converge pas vers une quantité déterministe quand  $n$  tend vers l'infini ; elle est elle-même distribuée selon une loi de Cauchy.

### A.4.8 Loi de Weibull

Paramètres :  $(\alpha, \lambda)$

$$\begin{aligned} f(x) &= \alpha \lambda^\alpha (x - \delta)^{\alpha-1} \exp[-(\lambda(x - \delta))^\alpha] \text{ pour } x > \delta, \\ F(x) &= 1 - \exp[-(\lambda(x - \delta))^\alpha]. \end{aligned}$$

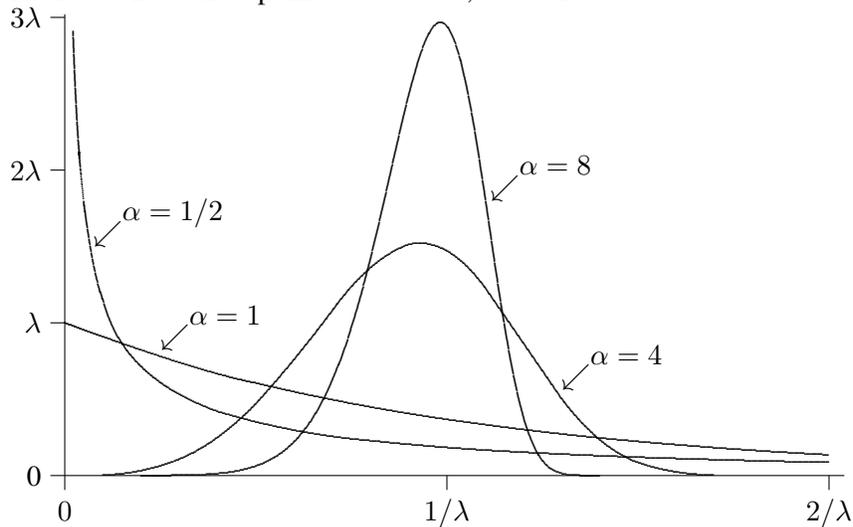
Paramètres :  $\delta \in \mathcal{R}$  (localisation),  $\alpha > 0$  (forme),  $\lambda > 0$  (échelle).

Souvent utilisé (avec  $\delta = 0$ ) pour modéliser les durées de vie.

Pour  $\alpha = 1$  : loi exponentielle.

Pour  $\alpha > 1$  : taux de panne *croissant*,  $CV > 1$ .

Pour  $\alpha < 1$  : taux de panne *décroissant*,  $CV < 1$ .



*Motivation* : le *minimum* de plusieurs v.a. i.i.d. suit approx. une loi de Weibull.

*Interprétation* : la panne d'un système survient lors de l'occurrence du premier événement parmi plusieurs événements indépendants.

### A.4.9 Loi de Gumbel

$(\alpha, \lambda)$

La loi de est aussi appelée loi de la valeur extrême (de type I).

$Y \sim \text{Weibull}(\alpha, \lambda)$  ssi  $X = \ln Y \sim \text{Gumbel}(\alpha, \lambda)$ .

La loi de Gumbel est populaire en théorie des choix discrets ; la différence de deux Gumbel indépendantes suit une distribution logistique.

**A.4.10 Loi beta**

La distribution *beta* de paramètres de forme  $\alpha > 0$  et  $\beta > 0$ , sur l'intervalle  $(0, 1)$ , a la densité

$$f(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \quad \text{for } 0 < x < 1.$$

Elle est parfois utilisée pour modéliser une proportion aléatoire. Sa moyenne et sa variance sont  $E[X] = \alpha/(\alpha + \beta)$  et  $\text{Var}[X] = \alpha\beta/[(\alpha + \beta)^2(\alpha + \beta + 1)]$ . Certaines densités de beta sont illustrées sur la Figure A.2 pour le cas symétrique ( $\alpha = \beta$ ) et A.3 pour le cas non-symétrique. For the symmetric case, all the density becomes concentrated at 1/2 (asymptotically) when  $\alpha \rightarrow \infty$  and split half and half between 0 and 1 when  $\alpha \rightarrow 0$ .

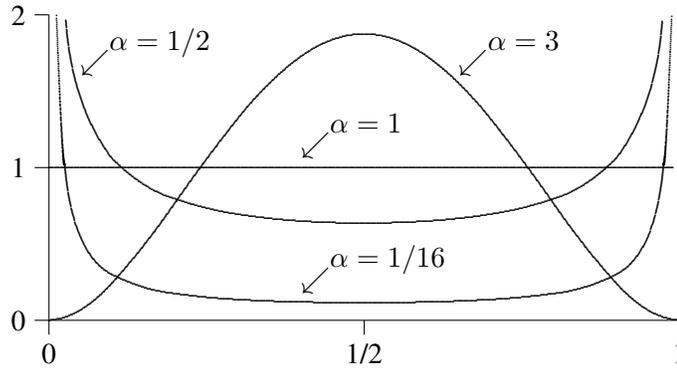


FIGURE A.2 – The symmetric beta density with parameters  $\alpha = \beta = 1/16, 1/2, 1,$  and 3.

On a  $X \sim \text{Beta}(\alpha, \beta)$  si et seulement si  $1 - X \sim \text{Beta}(\beta, \alpha)$ , ou encore si et seulement si  $X/(1 - X) \sim \text{Pearson-6}(\alpha, \beta, 1)$ . The  $\text{Beta}(1, 1)$  and  $U(0, 1)$  distributions are the same. The  $\text{Beta}(1, 2)$  and  $\text{Beta}(2, 1)$  are special cases of the distribution triangulaire. A beta random variable  $X$  over  $(0, 1)$  can be transformed into a beta random variable  $Y$  with the same shape of distribution, but over an arbitrary interval  $(a, b)$ , via the linear transformation  $Y = a + (b - a)X$ .

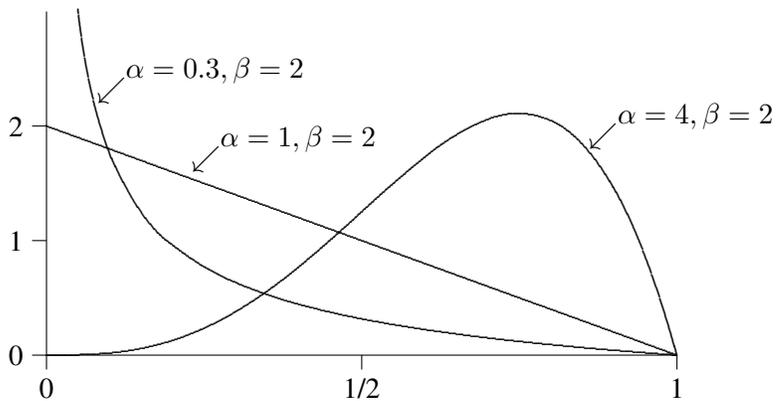


FIGURE A.3 – The beta density with parameters  $\beta = 2$  and  $\alpha = 0.3, 1,$  and  $4$ .

We have the following relationship between the binomial and beta distribution functions : If  $Y \sim \text{Binomiale}(n, p)$ , then  $P[Y \geq y] = P[X \leq p]$  where  $X \sim \text{Beta}(y, n-y+1)$ . This is sometimes used to approximate the binomial distribution function.

#### A.4.11 Loi du chi-deux

Paramètre :  $(n)$

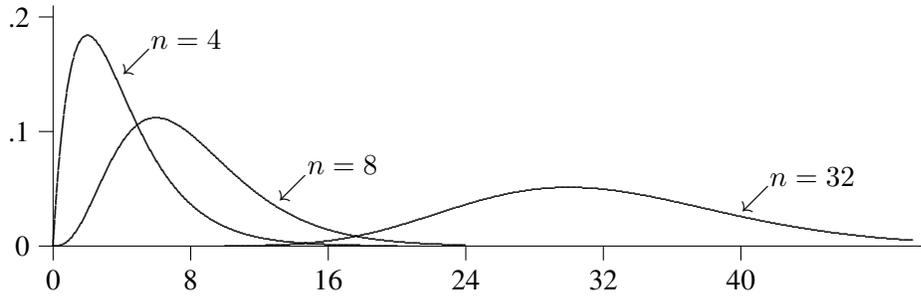
$$f(x) = \frac{x^{(n/2)-1} e^{-x/2}}{2^{n/2} \Gamma(n/2)} \text{ pour } x > 0.$$

Cas particulier de la loi gamma.

$E[X] = n$  et  $\text{Var}[X] = 2n$ .

Si  $X_1, \dots, X_n$  i.i.d.  $N(0, 1)$ , alors  $X = X_1^2 + \dots + X_n^2 \sim \text{chi-deux}(n)$ .

Utilisé dans les tests de chi-deux et pour calculer un intervalle de confiance pour la variance.



### A.4.12 Familles de Johnson

Très flexibles, grande variété de formes.  
 Peut imiter la plupart des lois standard.  
 Forme générale :

$$F(x) = \Phi[\gamma + \delta g((x - \xi)/\lambda)], \text{ for } x \in \mathcal{R}.$$

Paramètres :  $\gamma$  et  $\delta > 0$  (forme),  $\xi$  (localisation),  $\lambda > 0$  (échelle).  
 Transformation  $g$  choisie parmi :

$$g(y) = \begin{cases} \ln y & \text{famille lognormale,} \\ \sinh^{-1}(y) = \ln(y + \sqrt{y^2 + 1}) & \text{famille non bornée,} \\ \ln(y/(1 - y)) & \text{famille bornée,} \\ y & \text{famille normale.} \end{cases}$$

Dans chaque cas,  $Z = \gamma + \delta g((X - \xi)/\lambda) \sim N(0, 1)$ .  
 Ainsi  $X = \xi + \lambda g^{-1}((Z - \gamma)/\delta)$  où

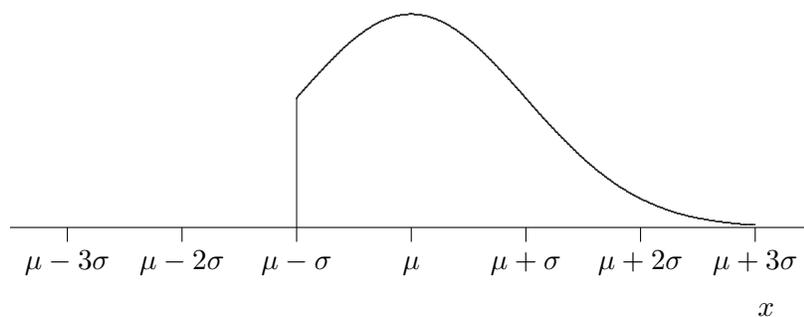
$$g^{-1}(z) = \begin{cases} e^z & \text{pour la famille lognormale,} \\ (e^z - e^{-z})/2 & \text{pour la famille non bornée,} \\ 1/(1 + e^{-z}) & \text{pour la famille bornée,} \\ z & \text{pour la famille normale.} \end{cases}$$

Densité :

$$f(x) = \frac{\delta}{\lambda\sqrt{2\pi}} f'((x - \xi)/\lambda) \exp \left[ -(\gamma + \delta f((x - \xi)/\lambda))^2/2 \right] \text{ pour } x \in H,$$

$H = (-\infty, \infty)$  pour normale et non bornée,  
 $H = [\xi, \xi + \lambda]$  pour bornée,  
 $H = [\xi, \infty)$  pour lognormale.

## A.5 Lois tronquées



Pour tronquer  $f(x)$  sur un intervalle  $(a, b)$  :

$$\tilde{f}(x) = \begin{cases} f(x)/K & \text{pour } a \leq x \leq b, \\ 0 & \text{ailleurs,} \end{cases}$$

où  $K = \int_a^b f(u) du$ .

Pour générer des v.a. selon  $\tilde{f}$  :  $U \sim \text{Uniforme}(F(a), F(b))$  et  $X = F^{-1}(U)$ .

## A.6 Lois décalées

$X = X_0 + a$  où  $X_0$  a une loi standard.

Exemples : exponentielle, gamma, lognormale, ...

Estimer le paramètre  $a$  à partir des données est parfois difficile !

## A.7 Mélanges de lois

$$f(x) = \sum_{j=1}^{\infty} w_j f_j(x),$$

où chaque  $f_j$  est une densité et les poids  $w_j$  somment à 1.

Population divisée en sous-ensembles, chaque sous-ensemble a sa propre distribution.

Mélange non dénombrable :

$$f(x) = \int_{\Theta} f_{\theta}(x)w(\theta)d\theta,$$

où  $\Theta \subseteq \mathcal{R}^d$ , et  $w$  une densité sur  $\Theta$ .

Pour générer  $X$ , générer d'abord  $Y$  selon les poids, puis  $X$  sous la densité  $f_Y$ .

Exemple :  $X \sim$  binomiale( $n, p$ ) où  $p \sim$  beta( $\alpha, \beta$ ).

## A.8 Lois multivariées

Un vecteur  $\mathbf{X} = (X_1, \dots, X_d)^T$  a une *loi multivariée* de fonction de répartition  $F$  si pour tout  $\mathbf{x} = (x_1, \dots, x_d)^T \in \mathcal{R}^d$ , on a

$$F(\mathbf{x}) = P[\mathbf{X} \leq \mathbf{x}] = P[X_1 \leq x_1, \dots, X_d \leq x_d].$$

On définira également les lois marginales en considérant la fonction de répartition associée à chaque composante du vecteur aléatoire :

$$\begin{aligned} F_j(x) &= P[X_j \leq x] \\ &= P[X_1 \leq +\infty, \dots, X_{j-1} \leq +\infty, X_j \leq x, X_{j+1} \leq x_{j+1}, \dots, X_d \leq +\infty], \end{aligned}$$

pour  $j = 1, \dots, d$ .

Deux variables  $X$  et  $Y$  sont jointement continues s'il existe une fonction nonnégative  $f(x, y)$ , appelée fonction de densité jointes de  $X$  et  $Y$ , telle que pour tous ensembles réels  $A$  et  $B$ ,

$$P[X \in A \cap Y \in B] = \int_B \int_A f(x, y) dx dy.$$

Le concept s'étend directement à plus de deux variables.

### A.8.1 Loi normale multivariée

Un des exemples les plus importants de lois multivariées est la loi normale multivariée ou multinormale. Un vecteur aléatoire  $\mathbf{X} = (X_1, \dots, X_d)^T$  suit une

normale multivariée  $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  s'il a pour densité

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} e^{-(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})/2}$$

où  $\mathbf{x} = (x_1, \dots, x_d)^T \in \mathcal{R}^d$ ,  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_d)^T \in \mathcal{R}^d$  (moyenne) et  $\boldsymbol{\Sigma} = \text{Cov}(\mathbf{X})$  (matrice de covariance). Dans ce cas,

$$\begin{aligned} E[X_j] &= \mu_j, \\ \text{Cov}(X_j, X_k) &= \sigma_{jk}. \end{aligned}$$

$X$  suit une loi *normale multivariée standard* si  $X \sim N(\mathbf{0}, \mathbf{I})$ , i.e. le vecteur moyenne est le vecteur nulle et la matrice de variance-covariance correspond à l'identité. Si  $\mathbf{X} \sim N(\mathbf{0}, \mathbf{I})$  et  $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$  (en d'autres termes,  $L$  est le facteur de Cholesky de  $\boldsymbol{\Sigma}$ ), alors  $\mathbf{Y} = \mathbf{L}\mathbf{X} + \boldsymbol{\mu} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ . Il est donc facile de générer  $\mathbf{Y}$  à partir de gaussiennes.

Dans le cas présent, la moyenne et la covariance suffisent pour décrire de manière univoque la distribution, mais ce n'est pas le cas en général pour les autres distributions.

Domaine	$\mathbb{R}^s$
Densité	$f_X(x_1, \dots, x_s) = \frac{1}{(2\pi)^{s/2}  \boldsymbol{\Sigma} ^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$

TABLE A.6 – Loi normale multivariée

## A.8.2 Indépendance, covariance, corrélation

$X_1, \dots, X_d$  sont *indépendantes* ssi  $F(x_1, \dots, x_d) = F_1(x_1) \cdots F_d(x_d)$ ; dans le cas contraire, les variables sont dites *dépendantes*. Il est possible de mesurer la dépendance entre deux variables aléatoires  $X$  et  $Y$  en spécifiant la covariance entre deux variables aléatoires :

$$\text{Cov}(X, Y) = E[XY] - E[X]E[Y], \quad (\text{A.1})$$

Si  $Y = X$ , (A.1) devient

$$\text{Cov}(X, X) = E[X^2] - E^2[X] = \text{Var}(X).$$

En supposant que  $X$  et  $Y$  ont des variances non nulles, nous pouvons également définir le coefficient de corrélation linéaire de Pearson :

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{(\text{Var}(X)\text{Var}(Y))^{1/2}} = \frac{E[XY] - E[X]E[Y]}{\sigma(X)\sigma(Y)}.$$

On a  $-1 \leq \rho(X, Y) \leq 1$ .

Si nous considérons plus de deux variables aléatoires, par exemples les composantes  $X_1, \dots, X_d$  du vecteur aléatoire  $\mathbf{X}$ , nous construirons la matrice de covariance

$$\Sigma = \text{Cov}[\mathbf{X}],$$

dont les éléments sont donnés par

$$\sigma_{ij} = \text{Cov}[X_i, X_j].$$

De la même manière, nous définirons la matrice de corrélation  $\mathbf{R}$ , formée des éléments  $\rho(X_i, X_j)$ . Contrainte de *validité* :  $\Sigma$  et  $\mathbf{R}$  doivent être *définies semi-positives*, i.e.,  $\mathbf{x}^T \Sigma \mathbf{x} \geq 0$  pour tout  $\mathbf{x} \in \mathcal{R}^d$ , et la diagonale de  $\mathbf{R}$  ne peut contenir que des 1.

Pour certains choix de marginales, il peut y avoir d'autres contraintes ! Les  $F_j$  peuvent provenir de différentes familles.

### A.8.3 Bornes de Fréchet

Il est parfois utile de chercher à minimiser ou à maximiser la corrélation entre deux variables aléatoires (voir Chapitre 7). Les résultats suivant permettent de dresser des pistes pour y parvenir.

**Theorème 15.** (e.g., Lehmann 1966)

Parmi les paires de v.a.  $(X, Y)$  dont les f.r. marginales sont  $F$  et  $G$ , la paire  $(X, Y) = (F^{-1}(U), G^{-1}(U))$  où  $U \sim U(0, 1)$ , maximise  $\rho[X, Y]$ , et la paire  $(X, Y) = (F^{-1}(U), G^{-1}(1 - U))$  minimise  $\rho[X, Y]$ .

La loi conjointe de corrélation *maximale* satisfait :

$$\begin{aligned} P\{X \leq x, Y \leq y\} &= P[F^{-1}(U) \leq x, G^{-1}(U) \leq y] \\ &= P[U \leq F(x), U \leq G(y)] \\ &= P[U \leq \min(F(x), G(y))] = \min(F(x), G(y)). \end{aligned}$$

Celle de corrélation *minimale* satisfait :

$$\begin{aligned} P\{X \leq x, Y \leq y\} &= P[F^{-1}(U) \leq x, G^{-1}(1 - U) \leq y] \\ &= P[U \leq F(x), 1 - U \leq G(y)] \\ &= P[1 - G(y) \leq U \leq F(x)] = \max(0, F(x) + G(y) - 1). \end{aligned}$$

**Exemple 36.** Soient  $X \sim \text{Normale}(\mu_1, \sigma_1^2)$  et  $Y \sim \text{Normale}(\mu_2, \sigma_2^2)$ . Soient  $U \sim U(0, 1)$  et  $Z = \Phi^{-1}(U)$ . *Corrélation maximale* :  $X = \mu_1 + \sigma_1 Z$  et  $Y = \mu_2 + \sigma_2 Z$ . On a

$$\rho(X, Y) = \frac{E[(X - \mu_1)(Y - \mu_2)]}{\sigma_1 \sigma_2} = \frac{E[\sigma_1 Z \sigma_2 Z]}{\sigma_1 \sigma_2} = E[Z^2] = 1.$$

*Corrélation minimale* : puisque  $\Phi^{-1}(1 - U) = -Z$ , c'est  $X = \mu_1 + \sigma_1 Z$  et  $Y = \mu_2 - \sigma_2 Z$ .

$$\rho(X, Y) = \frac{E[(X - \mu_1)(Y - \mu_2)]}{\sigma_1 \sigma_2} = \frac{E[\sigma_1 Z (-\sigma_2 Z)]}{\sigma_1 \sigma_2} = -E[Z^2] = -1.$$

Dans ce cas,  $\rho(X, Y)$  peut prendre n'importe quelle valeur dans  $[-1, 1]$ .

Ceci est vrai en général si  $X$  et  $Y$  ne diffèrent que par des paramètres de localisation et d'échelle.

#### A.8.4 Autres mesures de dépendance

Le coefficient de corrélation  $\rho(X, Y)$  dépend non seulement du niveau de dépendance entre  $(X, Y)$ , mais aussi de leurs lois marginales. Dans le cas continu, le coefficient de *corrélation des rangs* ou de *Spearman*,

$$\rho_s(X, Y) = \rho(F(X), G(Y)) = 12 E[F(X)G(Y)] - 3,$$

mesure la dépendance indépendamment des marginales. En effet, si  $X = F^{-1}(U_1)$  et  $Y = G^{-1}(U_2)$  pour des uniformes  $U_1$  et  $U_2$ , alors  $\rho_s(X, Y) = \rho(U_1, U_2)$ . Il peut prendre toutes les valeurs dans  $[-1, 1]$ , peu importe  $F$  et  $G$ , et existe toujours.

#### A.8.5 Dépendance vs corrélation.

Les coefficients  $\rho$  et  $\rho_s$  ne mesurent que la dépendance *linéaire*. Par exemple, si  $X \sim N(0, 1)$  et  $Y = X^2$ , il est clair que  $Y$  dépend de  $X$ , mais

$$\rho(X, Y) = \frac{E[XY] - E[X]E[Y]}{\sigma(X)\sigma(Y)} = E[X^3] - E[X]E[X^2] = 0.$$

L'indépendance implique l'absence de corrélation, mais pas l'inverse.

## A.9 Fonctions de dépendance (copules)

Soit  $\mathbf{X} = (X_1, \dots, X_d)$  v.a. continue, où  $P[X_j \leq x] = F_j(x)$ . Alors  $\mathbf{U} = (U_1, \dots, U_d) = (F_1(X_1), \dots, F_d(X_d))$  est un vecteur aléatoire dont les lois marginales sont  $U(0, 1)$ . Ce vecteur  $\mathbf{U}$  a une certaine fonction de répartition, disons  $C : P[U_1 \leq u_1, \dots, U_d \leq u_d] = C(u_1, \dots, u_d)$ .

En principe, on peut générer  $\mathbf{U}$  selon  $C$  et poser  $X_j = F_j^{-1}(U_j)$ . La fonction  $C$  spécifie la dépendance entre les  $U_j$ , indépendamment des marginales  $F_j$ . Une telle fonction de répartition  $C$ , dont les lois marginales sont toutes  $U(0, 1)$ , s'appelle une *copule* ou *fonction de dépendance* pour  $\mathbf{U}$ .

**Theorème 16** (Théorème de Sklar). *La loi de  $\mathbf{X}$  est spécifiée de façon unique par une copule  $C$  et les marginales  $F_j$ . On a*

$$F(x_1, \dots, x_d) = C(u_1, \dots, u_d) = C(F_1(x_1), \dots, F_d(x_d))$$

où  $u_j = F_j(x_j)$ . De plus,  $X_1, \dots, X_d$  sont indépendantes ssi  $C(u_1, \dots, u_d) = u_1 \cdots u_d$ .

Comment choisir  $C$ ? Trop de possibilités! On veut quelque chose de pratique à utiliser. Certaines méthodes *restreignent* volontairement le champs de possibilités pour  $C$ . En général, on peut choisir une f.r.  $G$  à  $d$  dimensions, de marginales  $G_j$ , et définir  $C$  par

$$C(u_1, \dots, u_d) = G(G_1^{-1}(u_1), \dots, G_d^{-1}(u_d)).$$

Une fois cette copule définie, on peut l'utiliser pour spécifier la dépendance pour un vecteur  $(X_1, \dots, X_d)$  de lois marginales  $F_j$ .

Pour générer  $(X_1, \dots, X_d)$ , on génère d'abord  $(Y_1, \dots, Y_d)$  selon  $G$ , on pose  $U_j = G_j(Y_j)$ , puis  $X_j = F_j^{-1}(U_j)$ , pour tout  $j$ .

## A.10 Lois empiriques et quasi-empiriques

Supposons que nous disposons des observations  $x_1, \dots, x_n$ , que nous renottons  $x_{(1)}, \dots, x_{(n)}$  après les avoir triées par ordre croissant. La fonction de répartition empirique est définie en partant de 0 et en augmentant sa valeur de  $1/n$  à chaque observation rencontrée :

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^n I[x_i \leq x].$$

Schématiquement, nous pouvons la représenter comme sur la Figure A.4.

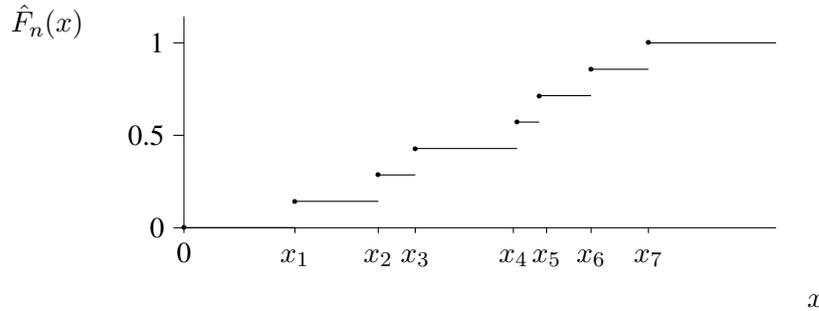


FIGURE A.4 – Fonction de répartition empirique

Générer des valeurs selon  $\hat{F}_n$  correspond à tirer des valeurs au hasard dans l'échantillon, avec remplacement. Si  $x_1, \dots, x_n$  sont les valeurs de variables aléatoires i.i.d. de répartition  $F$ , alors nous avons le résultat suivant

**Theorème 17.** *Convergence de la fonction de répartition empirique.*

$$D_n = \sup_{-\infty < x < \infty} |\hat{F}_n(x) - F(x)| \rightarrow 0 \text{ a.p.1 lorsque } n \rightarrow \infty,$$

et  $D_n$  converge en probabilité en  $O\left(\frac{1}{\sqrt{n}}\right)$ .

Par conséquent, pour  $n$  grand,  $F_n$  est une bonne approximation de  $F$ , et nous pouvons donc générer des valeurs directement de  $\hat{F}_n$  au lieu d'estimer une loi paramétrique. Nous ne pourrions toutefois générer que les mêmes valeurs qui sont dans l'échantillon. Une solution potentielle consiste à lisser  $\hat{F}_n$ . Le variante continue linéaire par morceaux, représentée sur la Figure A.5, s'exprime comme suit :

$$\tilde{F}_n(x) = \begin{cases} 0 & \text{if } x \leq x_{(1)}, \\ \frac{i-1}{n-1} + \frac{x-x_{(i)}}{(n-1)(x_{(i+1)}-x_{(i)})} & \text{if } x_{(i)} \leq x \leq x_{(i+1)}, \\ 1 & \text{if } x \geq x_{(n)}. \end{cases}$$

Il est ainsi possible de générer une valeur quelconque dans l'intervalle  $(x_{(1)}, x_{(n)})$ . Si on veut pouvoir générer des valeurs en dehors de  $(x_{(1)}, x_{(n)})$ , par exemple sur  $[0, \infty)$  pour une durée de vie, on peut ajouter une queue à la distribution. Par exemple, Bratley, Fox, and Schrage (1987) proposent une fonction linéaire par

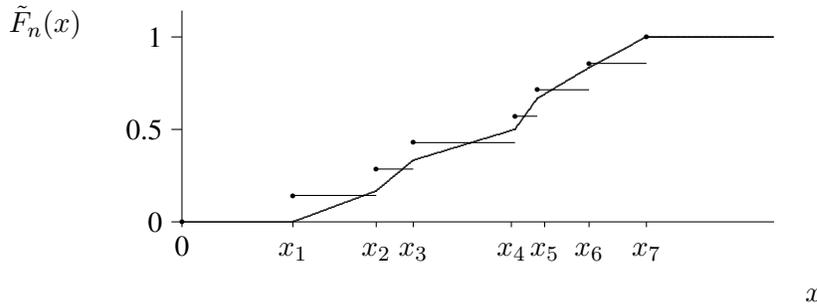


FIGURE A.5 – Fonction de répartition empirique lissée

morceaux jusqu'à  $x_{(n-k)}$ , avec une queue exponentielle par la suite, ajustée pour obtenir la même moyenne que dans les données :

$$\check{F}_n(x) = \begin{cases} \frac{i}{n} + \frac{x - x(i)}{(x_{(i+1)} - x(i))n} & \text{si } x(i) \leq x \leq x_{(i+1)}, 0 \leq i < n - k, \\ 1 - \frac{k}{n} e^{-\frac{x - x_{(n-k)}}{\theta}} & \text{si } x > x_{(n-k)}, \end{cases}$$

où  $x_{(0)} = 0$ ,  $k$  est un entier de 1 à 5 (disons), et

$$\theta = \frac{1}{k} \left( \frac{x_{(n-k)}}{2} + \sum_{i=n-k+1}^n (x(i) - x_{(n-k)}) \right).$$

Nous pourrions aussi ajuster la queue d'une autre loi (gamma, lognormale, etc.).

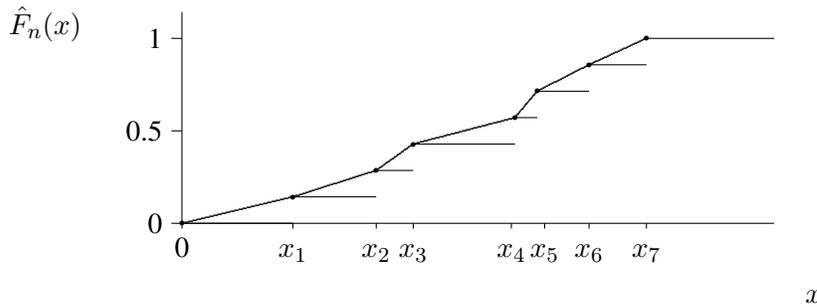


FIGURE A.6 – Fonction de répartition empirique prolongée

## A.11 Processus stochastiques

Un *processus stochastique* est une famille  $\{Y_t, t \in I\}$  de variables aléatoires définies sur un même espace de probabilité. L'indice  $t$  est souvent interprété comme le *temps*. Le processus est en *temps continu* si  $I$  est continu (e.g.,  $I = [0, \infty)$ ), et en *temps discret* si  $I$  est discret (e.g.,  $I = \{0, 1, 2, \dots\}$ ). Lorsque  $t$  est continu, on note souvent  $Y_t$  par  $Y(t)$ . On supposera ici que  $Y_t$  prend ses valeurs dans  $\mathcal{R}^d$ .

Afin de pouvoir appliquer des techniques d'inférences à partir de sorties d'une simulation, on fait souvent des hypothèses sur le processus stochastique sous-jacent, qui peuvent ne pas être vérifiées en pratique, mais visant à simplifier, voire rendre possible, l'analyse statistique des résultats. En particulier, pour un processus stochastique  $\{Y_t, t \geq 0\}$ , définissons :

- $\mu_t = E[Y_t]$ , la *fonction de moyenne*,
- $\sigma_t^2 = \text{Var}[Y_t]$ , la *fonction de variance*,
- $\text{Cov}[Y_t, Y_s]$ , la *fonction d'autocovariance*,
- $\rho_{t,s} = \text{Cov}[Y_t, Y_s] / \sigma_t \sigma_s$ , la *fonction d'autocorrelation*.

Nous introduisons deux concepts de stationnarité .

**Definition A.2:** Stationnarité faible

le processus est *faiblement stationnaire*, ou *stationnaire en covariance* si  $\mu_t \equiv \mu$ ,  $\sigma_t^2 \equiv \sigma^2$  et  $\rho_{t,t+s} \equiv \rho_s$  (autocorrélation de *délai*  $s$ ) sont indépendants de  $t$ .

**Definition A.3:** Stationnarité stricte

Le processus  $\{Y_t, t \geq 0\}$  est *strictement stationnaire* si pour tout vecteur  $(t_1, \dots, t_k)$  fixé, la loi conjointe de  $(Y_{t+t_1}, \dots, Y_{t+t_k})$  ne dépend pas de  $t$ .

Au contraire de la stationnarité stricte, la *loi* de  $Y_t$  peut dépendre de  $t$  dans le cas d'un processus faiblement stationnaire. En pratique, il est aussi très difficile de modéliser la loi conjointe d'un processus strictement stationnaire. Souvent, on se restreint à une classe étroite de modèles pour lesquels il ne reste à spécifier que quelques paramètres (e.g., moyenne, variance et autocorrélations).

### A.11.1 Processus markoviens

Un processus stochastique est *Markovien* si, conditionnellement à sa valeur présente au temps  $t$ , son évolution future est indépendante de son passé. Plus précisément, pour toute variable aléatoire  $X$  fonction de  $\{Y_s, s > t\}$ , la loi de  $X$  conditionnelle à  $\{Y_s, s \leq t\}$ , est la même que celle conditionnelle à  $Y_t$ . En d'autres termes,  $Y_t$  contient toujours assez d'information pour "générer" le futur. Nous parlerons de *chaîne de Markov temps discret* lorsque  $I = \{0, 1, \dots\}$ .

**Exemple 37** (File  $GI/G/1$ ).  $\{W_i, i \geq 0\}$  est markovien, car, en vertu de l'équation de Lindley,  $W_{i+1} = \max(0, W_i + S_i - A_i)$ .

$\{Q(t), t \geq 0\}$  n'est pas markovien, sauf pour une file  $M/M/1$ . En effet, le temps d'attente dépend du temps écoulé pour la personne en train d'être servie. Ce n'est plus le cas avec des variables exponentielles, vu leur propriété sans mémoire.

Posons :

- $T$ , l'horizon de la simulation ;
- $t_i$ , instant du  $i$ -ième événement  $e_i$  ;
- $Q_i$ , nombre de clients dans la file juste après  $e_i$  ;
- $\zeta_i$ , temps jusqu'à la prochaine arrivée ;
- $\xi_i$ , la durée résiduelle de service ( $-1$  si personne n'est servi) ;
- $\mathcal{S}_i = (t_i, Q_i, \zeta_i, \xi_i)$ , état de la simulation.

Le processus  $\{\mathcal{S}_i, i \geq 0\}$  est une chaîne de Markov.

En général,  $\{\mathcal{S}_i, i = 0, 1, 2, \dots\}$ , où  $\mathcal{S}_i$  est l'état du modèle de simulation au temps  $t_i$ , juste après l'événement  $e_i$ , est une chaîne de Markov. Par contre, le processus  $\{\mathcal{S}(t), t \geq 0\}$  défini par  $\mathcal{S}(t) = \mathcal{S}_{N(t)}$ , où  $N(t) = \sup\{i \mid t_i \leq t\}$  est le nombre d'événements durant  $(0, t]$ , ne l'est pas. Le processus  $\{\mathcal{S}(t), t \geq 0\}$  est uniquement déterminé par  $\{\mathcal{S}_i, i = 0, 1, \dots\}$ , mais pas l'inverse. Il peut y avoir des événements simultanés.

Pour les chaînes de Markov, on peut calculer les mesures de performance numériquement si le nombre d'états n'est pas trop grand.

### A.11.2 Filtrages et temps d'arrêt

Pour un processus  $\{Y_i, i = 0, 1, \dots\}$ ,  $\mathcal{F}_t = \sigma(Y_0, \dots, Y_t)$  représente toute l'information que l'on peut déduire en observant le processus jusqu'à l'étape  $t$ . De même, pour  $\{Y_t, t \geq 0\}$ ,  $\mathcal{F}_t$  représente l'information que l'on peut déduire de  $\{Y_s, 0 \leq s \leq t\}$ . La famille  $\{\mathcal{F}_t, t \geq 0\}$  s'appelle un *filtrage*.

Une variable aléatoire est  $\mathcal{F}_t$ -mesurable si on peut toujours déduire sa valeur de  $\mathcal{F}_t$ . Un temps d'arrêt par rapport à  $\{\mathcal{F}_t, t \geq 0\}$  est une variable aléatoire  $T$  (discrète ou continue), telle que  $T$  est  $\mathcal{F}_T$ -mesurable, c'est-à-dire si la valeur de  $T$  est connue au moment où elle est atteinte.

**Exemple 38.** *Centre d'appels*

Soient  $\{\mathcal{F}_i, i \geq 0\}$  le filtrage générée par  $\{\mathcal{S}_i, i \geq 0\}$ ,  $N$ , le nombre d'événements durant la journée et  $N'$ , le numéro d'événement de la dernière arrivée.  $N$  est un temps d'arrêt par rapport à  $\{\mathcal{F}_i, i \geq 0\}$ , mais pas  $N'$ .

### A.11.3 Processus de Poisson

Un processus de *comptage* est un processus en temps continu  $\{N(t), t \geq 0\}$ , à valeurs dans  $\{0, 1, 2, \dots\}$ , et dont les trajectoires sont non décroissantes et continues à droite<sup>2</sup>. Habituellement,  $N(0) = 0$ , et les instants de saut  $0 < T_1 \leq T_2 \leq \dots \leq T_j \leq \dots$  s'appellent les instants d'arrivées. Ainsi,  $N(t)$  représente le nombre d'arrivées durant  $[0, t]$ , on note  $A_j = T_j - T_{j-1}$ . Si les  $A_j$  sont des v.a. i.i.d., on a un *processus de renouvellement*.

On a un *processus de Poisson* si  $N(0) = 0$  et

- (a) les arrivées se font une à une (la prob. de 2 arrivées simultanées est nulle) ;
- (b) pour  $s, t \geq 0$ , la v.a.  $N(t+s) - N(t)$  est indépendante de  $\{N(u), u \leq t\}$  (i.e., elle ne dépend pas de l'histoire passée).

*Intuition* : un processus sera approximativement de Poisson si les événements arrivent au hasard, indépendamment les uns des autres.

Posons  $\Lambda(t) \stackrel{\text{def}}{=} E[N(t)]$ . Supposons que  $\Lambda(t)$  est continue partout, et dérivable sauf possiblement en un nombre fini de points sur tout intervalle fini. Alors

- $\lambda(t) = \Lambda'(t)$  est la *fonction de taux* du processus (aux points où la dérivée existe) ;
- $\Lambda(t) = \int_0^t \lambda(s) ds$  est la *fonction de taux cumulé*.

*Interprétation* : pour un petit  $\epsilon > 0$ , la probabilité d'un saut du processus dans l'intervalle de temps  $(t, t + \epsilon]$  est

$$\begin{aligned} P[N(t + \epsilon) - N(t) = 1] &\approx 1 - P[N(t + \epsilon) - N(t) = 0] \\ &\approx E[N(t + \epsilon)] - E[N(t)] \approx \lambda(t)\epsilon \end{aligned}$$

et la probabilité de plus d'un saut est  $o(\epsilon)$ .

2. Une fonction  $f$  est continue à droite en un point  $a$  si  $\lim_{x \rightarrow a, x > a} f(x) = f(a)$ .

Un processus de Poisson est dit *stationnaire* si  $\lambda(t) = \lambda > 0$  pour tout  $t \geq 0$ . Si  $\lambda = 1$ , on a un processus de Poisson *standard* (et donc  $\Lambda(t) = t$ ).

**Proposition 6.** *Si  $\{N(t), t \geq 0\}$  est un processus de Poisson et  $t_2 > t_1 \geq 0$ , alors  $N(t_2) - N(t_1)$  suit la loi de Poisson de moyenne*

$$\Lambda(t_2) - \Lambda(t_1) = \int_{t_1}^{t_2} \lambda(t) dt.$$

*Dans le cas stationnaire, la moyenne est  $(t_2 - t_1)\lambda$ .*

**Proposition 7.** *Un processus de comptage  $\{N(t), t \geq 0\}$ , avec  $N(0) = 0$ , est un processus de Poisson stationnaire de taux  $\lambda$  ssi les v.a.  $A_1, A_2, \dots$  sont i.i.d. exponentielles de paramètre  $\lambda$ .*

Dans le cas stationnaire, on peut donc générer les sauts en générant des *exponentielles* i.i.d..

On sait que pour un processus de Poisson, le *nombre d'arrivées* durant un intervalle  $(t_1, t_2]$  est  $\text{Poisson}(\Lambda(t_2) - \Lambda(t_1))$ . On peut donc générer ce nombre directement. Mais comment générer ensuite les *instants* de ces arrivées ? Dans le cas *stationnaire*, c'est facile :

**Proposition 8.** *Pour un processus de Poisson stationnaire, si  $n$  arrivées ont eu lieu durant un intervalle  $(t_1, t_2]$ , alors la loi conditionnelle des temps de ces  $n$  arrivées est la même que celle des statistiques d'ordre de  $n$  v.a. i.i.d.  $U(t_1, t_2)$  (i.e.  $n$  v.a. i.i.d.  $U(t_1, t_2)$ , triées par ordre croissant).*

## Standardisation

Un processus non-stationnaire peut être transformé en processus stationnaire simplement en étirant et comprimant l'échelle du temps. Soit  $X_j = \Lambda(T_j)$  pour  $j \geq 0$ ; définissons  $\{\tilde{N}(x), x \geq 0\}$  par

$$\tilde{N}(x) = N(\Lambda^{-1}(x)).$$

On a  $\tilde{N}(0) = 0$  et  $\tilde{N}(x)$  monte de 1 à chaque  $X_j$ .

**Proposition 9.**  *$\{\tilde{N}(x), x \geq 0\}$  est un processus de Poisson standard ssi  $\{N(t), t \geq 0\}$  est un processus de Poisson de taux cumulé  $\Lambda(\cdot)$ .*

*Démonstration.* Les axiomes (a) et (b) pour un processus de Poisson sont satisfaites pour  $\{\tilde{N}(t)\}$  ssi elles le sont pour  $\{N(t)\}$ , car  $\Lambda(\cdot)$  est continue et non-décroissante.

Reste à vérifier les fonctions de taux. Si  $E[N(t)] = \Lambda(t)$ , alors  $E[\tilde{N}(x)] = E[N(\Lambda^{-1}(x))] = \Lambda(\Lambda^{-1}(x)) = x$ . Inversement, si  $E[\tilde{N}(x)] = x$ , alors  $E[N(t)] = E[\tilde{N}(\Lambda(t))] = \Lambda(t)$ .  $\square$

Cette transformation permet de traiter et simuler un processus de Poisson quelconque, pourvu que l'on puisse calculer  $\Lambda^{-1}$  : pour *simuler*, on génère les  $X_j$ , puis on calcule les  $T_j = \Lambda^{-1}(X_j)$ .

*Superposition* de processus de Poisson : Si  $N_1(\cdot), \dots, N_k(\cdot)$  sont des processus de Poisson indépendants de taux  $\lambda_1(\cdot), \dots, \lambda_k(\cdot)$ , alors  $N(\cdot) = N_1(\cdot) + \dots + N_k(\cdot)$  est un processus de Poisson de taux  $\lambda(\cdot) = \lambda_1(\cdot) + \dots + \lambda_k(\cdot)$ .

*Décomposition* : inversement, si  $N(\cdot)$  est un processus de Poisson de taux  $\lambda(\cdot)$  et si chaque arrivée est de type  $j$  avec probabilité  $p_j$ , (indépendamment du temps et du passé), et  $N_j(t)$  est le nombre d'arrivées de type  $j$  durant  $(0, t]$ , alors  $N_j(\cdot)$  est un processus de Poisson de taux  $p_j\lambda(\cdot)$ .

Processus de Poisson *composé* : à chaque instant d'arrivée  $T_i$ ,  $N(\cdot)$  augmente (saute) de  $D_i$ , où les  $D_i$  sont i.i.d. et indépendants des  $T_i$  (arrivées en groupes).

Si  $\{\lambda(t), t \geq 0\}$  est lui-même un processus stochastique,  $\{N(t), t \geq 0\}$  devient un processus de Poisson *doublément stochastique*, ou processus de *Cox*. Dans ce cas,  $\{N(t), t \geq 0\}$  lui-même n'est un processus de Poisson que *conditionnellement* à  $\{\lambda(t), t \geq 0\}$ .

Exemples : centre d'appel, comptoir de glaces, etc.

#### A.11.4 Processus brownien

##### **Definition A.4:** Processus brownien

Un mouvement brownien avec paramètre de tendance  $\mu$  et paramètre de variance (ou coefficient de diffusion)  $\sigma^2$  est un processus  $\{X(t), t \geq 0\}$  tel que

- (a) si  $s \geq 0$  et  $t > 0$ , alors  $X(s+t) - X(s) \sim N(\mu t, \sigma^2 t)$  ;
- (b) pour tout ensemble d'intervalles disjoints  $(t_1, t_2], \dots, (t_{2k-1}, t_{2k}]$ , les incréments  $X(t_2) - X(t_1), \dots, X(t_{2k}) - X(t_{2k-1})$  sont mutuellement

indépendants.

Si  $\mu = 0$  et  $\sigma^2 = 1$ , nous parlerons de mouvement brownien standard. En pratique, tout processus brownien peut se ramener à un mouvement brownien standard vu que le processus standardisé

$$B(t) = [X(t) - \mu t]/(\sigma\sqrt{t})$$

est toujours un mouvement brownien standard. Pour  $s, t \geq 0$ , nous avons que  $\text{Cov}[X(s), X(t)] = \sigma^2 \min(s, t)$ .

D'autre part, si  $\mu$  peut varier en fonction du temps, et/ou  $\text{Cov}[X(s), X(t)]$  peuvent dépendre de  $s$  et  $t$ , on obtient un processus gaussien.

**Definition A.5:** Processus brownien géométrique

$\{S(t), t \geq 0\}$  est un mouvement brownien géométrique (GBM : geometric brownian motion) avec paramètre de dérive  $r = \mu + \sigma^2/2$  si  $\{X(t) = \ln S(t), t \geq 0\}$  est un mouvement brownien de paramètres  $\mu$  et  $\sigma^2$ .

En d'autres mots, si  $\{B(t), t \geq 0\}$  est un mouvement brownien standard avec  $B(0) = 0$ ,  $\{S(t), t \geq 0\}$  est un mouvement brownien géométrique si

$$S(t) = S(0) \exp[\mu t + \sigma B(t)].$$

Pour  $t_0 \geq 0$  et  $t > 0$ ,  $S(t_0 + t)/S(t_0)$  suit une loi lognormale( $\mu t, \sigma^2 t$ ). Par conséquent, sur un intervalle de temps donné, la variation relative ne dépend pas de la valeur initiale.

Un processus brownien géométrique peut aussi être spécifié au moyen de l'équation différentielle

$$\frac{dS(t)}{S(t)} = r dt + \sigma dB(t).$$

Un exemple courant d'utilisation en finance est le modèle de Black-Scholes.

**A.11.5 Estimation de moyennes, variances, et corrélations**

Supposons que  $X_1, X_2, \dots, X_n$  sont des variables i.i.d. de moyenne finie  $\mu$  et de variance  $\sigma^2$ . La moyenne échantillonnale

$$\hat{\mu}_n = \bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i.$$

est un estimateur non biaisé de  $\mu$ , c'est-à-dire  $E[\bar{X}_n] = \mu$ . Similairement, la variance échantillonnale

$$S_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2 = \frac{1}{n-1} \left( \sum_{i=1}^n X_i^2 - n(\bar{X}_n)^2 \right),$$

est un estimateur non biaisé de  $\sigma^2$ .

*Démonstration.* Nous avons

$$\begin{aligned} E[S_n^2] &= \frac{1}{n-1} E \left[ \sum_{i=1}^n (X_i - \bar{X}_n)^2 \right] \\ &= \frac{1}{n-1} E \left[ \sum_{i=1}^n (X_i^2 - 2X_i\bar{X}_n + \bar{X}_n^2) \right] \\ &= \frac{1}{n-1} E \left[ \sum_{i=1}^n \left( X_i^2 - \frac{2}{n} X_i \sum_{j=1}^n X_j + \frac{1}{n^2} \left( \sum_{j=1}^n X_j \right)^2 \right) \right] \\ &= \frac{1}{n-1} E \left[ \sum_{i=1}^n X_i^2 - \frac{2}{n} \sum_{i=1}^n \sum_{j=1}^n X_i X_j + \frac{n}{n^2} \sum_{i=1}^n \sum_{j=1}^n X_i X_j \right] \\ &= \frac{1}{n-1} E \left[ \sum_{i=1}^n X_i^2 - \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n X_i X_j \right] \\ &= \frac{1}{n-1} E \left[ \sum_{i=1}^n X_i^2 - n(\bar{X}_n)^2 \right], \end{aligned}$$

ou encore,

$$\begin{aligned}
 E[S_n^2] &= \frac{1}{n-1} E \left[ \sum_{i=1}^n X_i^2 - \frac{1}{n} \sum_{i=1}^n X_i^2 - \frac{1}{n} \sum_{i=1, j=1, i \neq j}^n X_i X_j \right] \\
 &= \frac{1}{n-1} E \left[ \frac{n-1}{n} \sum_{i=1}^n X_i^2 \right] - \frac{1}{n(n-1)} E \left[ \sum_{i=1, j=1, i \neq j}^n X_i X_j \right] \\
 &= E[X^2] - \frac{1}{n(n-1)} \sum_{i=1, j=1, i \neq j}^n E[X_i X_j],
 \end{aligned}$$

et par l'indépendance des  $X_i$ ,

$$\begin{aligned}
 E[S_n^2] &= E[X^2] - \frac{1}{n(n-1)} \sum_{i=1, j=1, i \neq j}^n E[X_i] E[X_j] \\
 &= E[X^2] - \frac{1}{n(n-1)} n(n-1) E[X]^2 \\
 &= E[X^2] - E[X]^2 \\
 &= \sigma^2.
 \end{aligned}$$

□

### A.11.6 Convergence

**Théorème 18.** *Supposons que  $\sigma^2 < \infty$ . Lorsque  $n \rightarrow \infty$  :*

- (i) Loi forte des grands nombres :  $\lim_{n \rightarrow \infty} \hat{\mu}_n = \mu$  avec probabilité 1.
- (ii) Théorème de la limite centrale :

$$\begin{aligned}
 \frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} &\Rightarrow N(0, 1), \text{ i.e.,} \\
 \lim_{n \rightarrow \infty} P \left[ \frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \leq x \right] &= \Phi(x) = P[Z \leq x]
 \end{aligned}$$

pour tout  $x \in \mathcal{R}$ , où  $Z \sim N(0, 1)$  et  $\Phi(\cdot)$  sa fonction de répartition.

Les deux propriétés tiennent aussi si on remplace  $\sigma^2$  par son estimateur sans biais  $S_n^2$ . Ainsi, pour  $n$  grand et un niveau de confiance  $1 - \alpha$ ,

$$P[|\hat{\mu}_n - \mu| \leq x S_n / \sqrt{n}] \approx \Phi(x) = 1 - \alpha.$$

Intervalle de confiance au niveau  $\alpha$  :

$$(\hat{\mu}_n \pm z_{\alpha/2} S_n / \sqrt{n}), \text{ où } z_{\alpha/2} = \Phi^{-1}(\alpha/2).$$

Exemple :  $z_{\alpha/2} \approx 1.96$  pour  $\alpha = 0.05$ .

La largeur de l'intervalle de confiance est asymptotiquement proportionnelle à  $\sigma/\sqrt{n}$ , donc converge en  $O(n^{-1/2})$ .

Si  $n$  est petit et les  $X_i$  suivent la loi normale, alors  $\sqrt{n}(\hat{\mu}_n - \mu)/S_n \sim Student(n - 1)$ .

Mais si la loi des  $X_i$  est très asymétrique, ces intervalles ne plus valides même comme approximations.

## A.12 Illustration avec SSJ

```
import umontreal.iro.lecuyer.rng.*;
import umontreal.iro.lecuyer.probdist.*;
import umontreal.iro.lecuyer.randvar.*;
import umontreal.iro.lecuyer.stat.*;

public class Nonuniform {
    // The parameter values are hardwired here to simplify the program.
    double lambda = 5.0;    double p = 0.2;
    double alpha = 2.0;    double beta = 1.0;
    double mu = 5.0;    double sigma = 1.0;

    RandomStream stream = new LFSR113();
    RandomVariateGenInt genN = new RandomVariateGenInt
        (stream, new PoissonDist (lambda));    // For N
    RandomVariateGen genY = new GammaAcceptanceRejectionGen
        (stream, new GammaDist (alpha, beta));    // For Y_j
    RandomVariateGen genW = new RandomVariateGen
        (stream, new LognormalDist (mu, sigma));    // For W_j

    // Generates and returns X.
    public double generateX () {
        int N; int M; int j; double X = 0.0;
        N = genN.nextInt();
        M = GeometricDist.inverseF (p, stream.nextDouble());    // Uses static method
        for (j = 0; j < N; j++) X += genY.nextDouble();
        for (j = 0; j < M; j++) X += genW.nextDouble();
        return X;
    }
}
```

```

// Performs n indep. runs and collects statistics in statX.
public void simulateRuns (int n) {
    TallyStore statX = new TallyStore (n);
    for (int i=0; i<n; i++) statX.add (generateX ());
    System.out.println (statX.report ());
    statX.quickSort ();
    double [] data = statX.getArray ();
    System.out.printf ("0.10 quantile: %9.3f%n", data [(int)(0.10 * n)]);
    System.out.printf ("0.50 quantile: %9.3f%n", data [(int)(0.50 * n)]);
    System.out.printf ("0.90 quantile: %9.3f%n", data [(int)(0.90 * n)]);
    System.out.printf ("0.99 quantile: %9.3f%n", data [(int)(0.99 * n)]);
}

public static void main (String [] args) {
    (new Nonuniform ()).simulateRuns (10000);
}
}

```

## A.13 Tests d'hypothèse

### A.13.1 Test d'adéquation

Soit  $\{x_1 \dots x_n\}$  un échantillon de  $n$  réalisations indépendantes de la v.a.  $X$ . Soit  $L(x)$  la loi de distribution inconnue de  $X$ . L'hypothèse  $H_0$  est que les données ont été tirées d'une distribution  $L^*(x)$ , i.e. :

$$\begin{aligned}
 H_0 &: L(x) = L^*(x), \\
 H_1 &: L(x) \neq L^*(x).
 \end{aligned}$$

Sous l'hypothèse  $H_0$ , on peut considérer que l'écart  $E_i - O_i$  entre distribution théorique et distribution empirique est distribué normalement. Dans ces conditions,  $I$  tend vers une loi du  $\chi^2$  à  $\nu = n - p$  degrés de liberté, où  $p$  est la réduction dans les degrés de liberté.  $p$  vaut ici  $s + 1$  où  $s$  est le nombre de paramètres de la distribution.

La région d'acceptation du test est l'intervalle  $(0, \chi_{\nu, 1-\alpha}^2)$  tel que la probabilité d'une variable du  $\chi^2$  à  $\nu$  degrés de liberté prenne une valeur dans cet intervalle soit au moins égale à  $1 - \alpha$  ( $\alpha$  étant l'erreur de première espèce relative au test). Si la valeur de l'indicateur est supérieure à  $\chi_{\nu, 1-\alpha}^2$ , on rejette  $H_0$ .

## A.14 Exercices

1. Montrez que si  $X \sim \text{Géométrique}(p)$ , alors  $P[X = y + x \mid X \geq y] = P[X = x]$ , où  $x$  et  $y$  sont naturels. Ceci est appelé la *propriété sans mémoire* de la distribution géométrique.
2. Montrez que si  $X_1, \dots, X_k$  sont des variables aléatoires exponentielles indépendantes, de taux  $\lambda_1, \dots, \lambda_k$  respectivement, alors  $X = \min(X_1, \dots, X_k)$  est exponentiel de taux  $\lambda = \lambda_1 + \dots + \lambda_k$ .
3. Vérifiez mathématiquement la formule relative au coefficient de corrélation de Spearman :

$$\rho_s(X, Y) = \rho(F(X), G(Y)) = 12 E[F(X)G(Y)] - 3,$$

où  $X$  et  $Y$  sont deux variables aléatoires de fonctions de répartition respectives  $F(x)$  et  $G(Y)$ .

4. Soit  $X_1, \dots, X_n$ ,  $n$  variables aléatoires i.i.d. Quelle est la distribution de la fonction indicatrice  $I(X_i \leq x)$  ? Cette fonction est définie comme suit :

$$I(X_i \leq x) = \begin{cases} 1 & \text{si } X_i \leq x, \\ 0 & \text{sinon.} \end{cases}$$

Partant de là, quelle est la moyenne et la variance de la fonction empirique construite à partir des observations  $x_1, \dots, x_n$  issues de  $X_1, \dots, X_n$  ?

5. Il est courant en sécurité routière de suggérer le maintien d'un écart de 2 secondes par rapport au véhicule qui précède. En considérant une distribution log-normale pour le temps de réaction, de paramètres  $\mu = 0.14$  et  $\sigma = 0.44$ , calculez à l'aide de SSJ le temps de réaction moyen, ainsi que les quantiles 0.90, 0.95. Sur base de ces résultats, considérez-vous la suggestion appropriée ? Considérons à présent l'utilisation d'une distribution normale, de moyenne 1.25 et d'écart-type 0.46. Répétez le calcul des quantiles 0.90 et 0.95. Quelle est la probabilité de générer un temps de réaction négatif ? Tronquez la distribution en 0 pour ne produire que des temps positifs. Que deviennent la moyenne, l'écart-type, les quantiles 0.9 et 0.95 ? Note : il est conseillé d'utiliser le paquetage `probdist` de SSJ



## Annexe B

# Rappels de géométrie et d'algèbre

### B.1 Corps

Un corps est un ensemble  $K$  muni de deux lois internes notées en général  $+$  et  $\times$  vérifiant

- $(K, +)$  forme un groupe commutatif dont l'élément neutre est noté  $0$  ;
- $(K \setminus \{0\}, \times)$  forme un groupe multiplicatif ;
- la multiplication est distributive pour l'addition (à gauche comme à droite), c'est-à-dire que

$$\forall (a, b, c) \in K^3, \quad a \times (b+c) = a \times b + a \times c \quad \text{et} \quad (b+c) \times a = b \times a + c \times a.$$

On parle alors du corps  $(K, +, \times)$

### B.2 Hyperplan

Un hyperplan est un concept géométrique généralisant en grande dimension les concepts de ligne dans la géométrie planaire euclidienne et un plan dans la géométrie euclidienne à 3 dimensions. Les types les plus familiers d'hyperplans sont les hyperplans affine et linéaire ; moins courants sont les hyperplans projectifs.

Dans un espace unidimensionnel (une ligne droite), un hyperplan est un point ; il divise une ligne en deux rayons. Dans un espace bidimensionnel (comme le plan  $xy$ ), un hyperplan est une ligne ; il divise le plan en deux demi-plans. Dans un espace

tridimensionnel, un hyperplan est un plan ordinaire. Il divise l'espace en deux demi-espaces. Ce concept peut aussi être appliqué à un espace à quatre dimensions ou plus, où l'objet divisant l'espace est simplement appelé un "hyperplan".

## Annexe C

# Systemes de file d'attente

Un système de file d'attente consiste d'un ou de plusieurs serveurs qui fournissent un service d'une certaine nature à des clients qui se présentent. Les clients qui arrivent et trouvent tous les serveurs occupés rejoindront (généralement) une ou plusieurs files (ou lignes) devant les serveurs, d'où la qualification de file d'attente. Quelques exemples de files d'attente sont repris dans la Table C. Le système de file

Systeme	Serveurs	Clients
Banque	Guichets	Clients
Hôpital	Docteurs, infirmières, lits	Patients
Systeme informatique	Unités centrales, dispositifs I/O	Travaux
Manufacture	Machines, travailleurs	Composants
Centre d'appel	Serveurs	Appels entrants

TABLE C.1 – Exemples de files d'attente

d'attente est caractérisé par trois composants : le processus d'arrivée, le mécanisme de service, la discipline de file.

Spécifier le **processus d'arrivée** pour un système de file d'attente revient à décrire comment les clients arrivent dans le système. Soit  $A_i$  le temps d'inter-arrivées entre les arrivées des  $(i - 1)^e$  et  $i^e$  clients. Si  $A_1, A_2, \dots$ , sont supposés être des variables i.i.d., nous dénoterons le temps d'interarrivée moyen (ou espéré) par  $E[A]$  et appellerons  $\lambda = 1/E[A]$  le taux d'arrivée des clients.

Le **mécanisme de service** pour un système de file d'attente est articulé en

spécifiant le nombre de serveurs (dénnoté par  $s$ ), où chaque serveur a sa propre file ou au contraire une seule file alimente tous les serveurs, et la distribution de probabilité des temps de service des clients. Soit  $S_i$  le temps de service du  $i^e$  client. Si  $S_1, \dots$  sont des variables aléatoires i.i.d., nous dénoterons le temps de service moyen d'un client par  $E[S]$  et appelleront  $\omega = 1/E[S]$  le taux de service d'un serveur.

La **discipline de file** réfère à la manière dont un serveur choisit le prochain client à partir d'une file (s'il en existe une) quand le serveur termine le service du client courant. Les disciplines courantes sont la règle FIFO (first-in, first-out : premier arrivé, premier servi), la règle LIFO (last-in, first-out : dernier arrivé, premier servi), la règle par priorités (les clients sont servis suivant leur importance, ou sur base des exigences de leur service).

## C.1 Notations pour les systèmes de files d'attentes

Certaines files d'attente se rencontrent si souvent en pratique que des notations standards ont été développées pour elles. Supposons en particulier que

1.  $s$  serveurs existent en parallèle, et une queue FIFO alimente tous les serveurs ;
2.  $A_1, A_2, \dots$  sont des variables aléatoires i.i.d ;
3.  $S_1, S_2, \dots$  sont des variables aléatoires i.i.d ;
4. les  $A_i$  et les  $S_i$  sont indépendants.

Un tel système sera dénoté une file  $GI/G/s$ , où  $GI$  (general independant) réfère à la distribution des  $A_i$  et  $G$  (general) réfère à la distribution des  $S_i$ . Si des distributions spécifiques sont données pour les  $A_i$  et les  $S_i$ , des symboles dénotant ces distributions sont utilisés en place de  $GI$  et de  $G$ . En particulier, le symbole  $M$  est utilisé pour la distribution exponentielle en raison de son caractère markovien, i.e. la propriété sans mémoire. La lettre  $D$  est employée pour des temps déterministes, i.e. constants.  $s$  désigne le nombre de serveurs. Dés lors, un système de file d'attente avec un seul serveur et des temps d'inter-arrivées et de service exponentiels, et une discipline de file FIFO est appelé file  $M/M/1$ .

Pour n'importe quelle file  $GI/G/s$ , nous appellerons la quantité

$$\rho = \frac{\lambda}{s\omega} \tag{C.1}$$

le facteur d'utilisation du système de file d'attente ( $s\omega$  est le taux de service quand tous les serveurs sont occupés). C'est une mesure de l'importance d'utilisation des ressources du système de file d'attente.

## C.2 Mesures de performance pour les files d'attente

Il existe de nombreuses mesures de performance possibles pour les systèmes de file d'attente ; parmi celles-ci, nous en retiendrons quatre fréquemment utilisées pour les études mathématiques de tels systèmes. Posons

- $W_i$  : délai dans la file du  $i^e$  client ;
- $D_i = W_i + S_i$  : temps d'attente dans le système du  $i^e$  client ;
- $q(t)$  : nombre de clients dans la file au temps  $t$  ;
- $L(t)$  : nombre de clients dans le système au temps  $t$ , i.e.  $q(t)$  plus le nombre de clients en train d'être servis au temps  $t$ .

Les mesures

$$w = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n W_i}{n} \quad \text{a.p. 1,} \quad \text{et} \quad d = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n D_i}{n} \quad \text{a.p. 1,}$$

si elles existent, sont appelées délai moyen en état stable et temps d'attente moyen en état stable. Similairement, les mesures

$$q = \lim_{T \rightarrow \infty} \frac{\int_0^T q(t) dt}{T} \quad \text{a.p. 1,} \quad \text{et} \quad L = \lim_{T \rightarrow \infty} \frac{\int_0^T L(t) dt}{T} \quad \text{a.p. 1,}$$

si elles existent, sont appelées nombre moyen par unité de temps en état stable dans la file, et nombre moyen par unité de temps en état stable dans le système. Notons que  $\rho < 1$  est une condition nécessaire pour que  $d$ ,  $w$ ,  $q$  et  $L$  existent pour une file d'attente  $GI/G/1$ . Dans ce cas, si  $d$  et  $w$  existent, nous avons les équations de conservation

$$q = \lambda w \quad \text{et} \quad L = \lambda d. \quad (\text{C.2})$$

Une autre égalité d'intérêt pratique est donnée par

$$d = w + E[S],$$

et donc

$$q = L - \rho.$$

Dans le cas d'une file  $M/M/1$ , nous pouvons en outre montrer que le nombre moyen en état stable dans le système est donné par

$$L = \frac{\rho}{1 - \rho}.$$

Pour plus d'informations, le lecteur est invité à consulter notamment la Section 1B de Law [8].

**Exemple 39** (Un exemple de modèle mathématique : la loi de Little). *La loi de Little dit que le nombre moyen  $N$  de clients dans un système stable (ou sur un horizon infini), est égal à leur fréquence moyenne d'arrivée  $\lambda$ , multipliée par leur temps moyen  $T$  passé dans le système :*

$$N = \lambda T \tag{C.3}$$

*L'équation C.3 est valable sous des conditions assez générales. Encore faut-il que le système soit en état stable !*

# Bibliographie

- [1] Moshe Ben-Akiva et Steven R. Lerman. *Discrete Choice Analysis : Theory and Application to Travel Demand*. The MIT Press, 1985.
- [2] G. E. P. Box et Mervin E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2) :610–611, 1958.
- [3] Xiren Cao. Some common misconceptions about performance modeling and validation. *Performance Evaluation Review*, 21(2), 1993.
- [4] Luc Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, NY, USA, 1986.
- [5] U. Dieter. How to calculate shortest vectors in a lattice. *Mathematics of Computation*, 29(131) :827–833, 1975.
- [6] Bradley Efron et Robert J. Tibshirani. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA, 1993.
- [7] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, New York, NY, USA, 2004.
- [8] Averill M. Law. *Simulation Modeling & Analysis*. McGraw-Hill, Boston, MA, USA, 4th édition, 2007.
- [9] P. L'Ecuyer, M. Mandjes et B. Tuffin. Importance sampling and rare event simulation. Dans G. Rubino et B. Tuffin, editors, *Rare Event Simulation Using Monte Carlo Methods*, pages 17–38. Wiley, 2009. Chapter 2.
- [10] Derrick H. Lehmer. Mathematical methods in large-scale computing units. Dans *Proceedings of the Second Symposium on Large Scale Digital Computing Machinery*, pages 141–146, Cambridge, United Kingdom, 1951. Harvard University Press.
- [11] Theodore G. Lewis et William H. Payne. Generalized feedback shift register pseudorandom number algorithm. *Journal of the Association for Computing Machinery*, 20(3) :456–468, 1973.

- [12] Makoto Matsumoto et Takuji Nishimura. Mersenne twister : A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8(1) :3–30, 1998.
- [13] Stephen K. Park et Keith W. Miller. Random number generators : good ones are hard to find. *Communications of the ACM*, 31(10) :1192–1201, 1988.
- [14] David Revelt et Kenneth Train. Mixed logit with repeated choices. *Review of Economics and Statistics : Households' Choices of Appliance Efficiency Effect*, 80(4), 1998.
- [15] J.W. Schmidt et R.E. Taylor. *Simulation and Analysis of Industrial Systems*. Richard D. Irwin, Homewood, Illinois, USA, 1970.
- [16] Howard M. Taylor et Samuel Karlin. *An Introduction To Stochastic Modeling*. Academic Press, third edition édition, 1998.
- [17] Kenneth Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, New York, NY, USA, 2003.
- [18] John von Neumann. Various techniques used in connection with random digits. Dans A.S. Householder, G.E. Forsythe et H.H. Germond, editors, *Monte Carlo Method*, pages 36–38. National Bureau of Standards Applied Mathematics Series, 12, Washington, D.C. : U.S. Government Printing Office, 1951.
- [19] Michael J. Wichura. Algorithm as 241 : The percentage points of the normal distribution. *Applied Statistics*, 37 :477–484, 1988.
- [20] Brian L. Wiens. When log-normal and gamma models give different results : a case study. *American Statistical Association*, 53(2) :89–93, 1999.

# Index

- a.p.1, voir convergence avec probabilité 1
- acceptation-rejet, 75
- analyse de sensibilité, 15
  
- biais, 84, 96
- Box-Muller, 78
  
- chaîne de Markov, 160
- champs de Galois, 56
- Cholesky, 156
- composition, 74
- consistence, 85
- convergence, 138
  - avec probabilité 1, 139
  - presque sûre, voir avec probabilité 1
- copule, 156
- corps, 171
- corrélation, 154
  - de Spearman, 155
  - des rangs, 155
  - linéaire de Pearson, 154
  - matrice de, 154
- covariance, 153
  - matrice de, 154
- crédibilité, 11
  
- Delta
  - théorème, 97
- distribution
  - bernouilli, 139
  - Cauchy, 147
  - empirique, 93
  - Erlang, 145
  - exponentielle, 145
  - géométrique, 140
  - Gumbel, 21, 148
  - multinomiale, 139
  - normale, 71, 142
    - multivariée, 21
  - Poisson, 141
  - triangulaire, 71, 142
  - Weibull, 147
  
- efficacité, 84, 113
- erreur
  - absolue, 84
  - de couverture, 85
  - quadratique moyenne, 84
  - relative, 84
- espérance, 138
- estimateur, 19, 84
- état, 1, 25
- événement, 25
- événements discrets, 5, 25
  
- facteur
  - d'utilisation, 16, 175
- file
  - d'attente, 173
  - $GI/G/1$ , 27, 160
  - $GI/G/s$ , 174
  - $M/G/1$ , 16
  - $M/M/1$ , 174
- filtrage, 161
- fonction
  - indicatrice, 21
  - fonction de répartition, 137
    - empirique, 157
  - fonction de survie, 137
- formule
  - Pollaczek-Khintchine, 16
  
- générateur
  - récuratif multiple, 44

- Standard Minimal, 41
- $GI/G/1$ , 107
- graphe
  - histogramme-densité, 23
- Greenshields, 4
- histogramme, 23
- horloge de simulation, voir simulation
- hyperplan, 50, 52, 171
- IIA, 21
- indépendance, 153
- intervalle de confiance, 85
- inversion, 70
- lacunaire, 53
- LFSR, 57
- limite centrale, 88, 144, 166
- Lindley, 31, 121, 160
- logit, 21
- loi de Little, 176
- marginales, 153
- modèle, 1, 3
- plug-in, 93
- Poisson, 77
- polynôme caractéristique, 57
- probabilité
  - espace, 135
  - mesure, 136
- probit, 21
- processus, 32
  - brownien, 164
    - géométrique, 164
    - standard, 164
  - de comptage, 161
  - markovien, 160
  - de Poisson, 77, 161
  - régénératif, 106
  - de renouvellement, 161
  - stationnaire
    - faiblement, 101
  - stochastique, 159
- quantile
  - $q$ -quantile, 91
- quasi-Monte Carlo, 132
- racine primitive, 40
- réduction de variance
  - facteur, 85
- simulation, 2
  - horloge de, 27
- Sklar
  - théorème de, 156
- SSJ, 8, 33
- stationnarité, 15, 159
  - en covariance, voir stationnarité faible
  - faible, 160
  - stricte, 160
- système, 1
- taux de service, 175
- temps
  - d'arrêt, 161
  - continu, 159
  - discret, 159
- temps de la simulation, 25
- v.a., voir variable aléatoire
- valeur à risque, 92
- validation, 11
- variable aléatoire, 136
- variance, 84
- vérification, 11
- vraisemblance
  - fonction de, 19
  - maximum de, 19
- Welch, 100
  - heuristique, 103