Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Mouloud Mammeri, Tizi-Ouzou

Faculté des Sciences

Département: Mathématiques

Mémoire de Master

en
Mathématiques appliquées

Option: Processus aléatoires et statistique de la décision

Thème:

Sur les Variables de Contrôle dans les méthodes de Monte Carlo

---

Présenté par

Hand Bersi

Devant le jury d'examen composé de:

Mr Mohand Arezki Boudiba	Maître de Conférences A	UMMTO	Président
Mr Hocine Fellag	Professeur	UMMTO	Rapporteur
Melle Lynda Atil	Maître de Conférences B	UMMTO	Examinatrice

Année 2013

Remerciements

Je tiens avant tout à remercier mon promoteur monsieur **Hocine Fellag**, qui a été d'un immense soutien pendant toute la durée de ce mémoire. Grâce à sa perspicacité et à son expertise remarquables, il a été une grande source d'inspiration et de conseils.

J'aimerais aussi remercier les membres du jury et mes enseignants, monsieur **Mohand Arezki Boudiba** et Mlle **Atil Lynda** d'avoir accepté de faire partie du jury, et pour le temps pris pour lire et apprécier ce travail.

" Ce qui dit : j'arrive, arrivera, inutile d'anticiper. Les drames ne gâchent jamais les précieux moments de la vie..., quand le drame est là, on le vit comme une étape, comme une fin. Le héros parmi les humains n'est pas celui qui ne tombe jamais, mais celui qui tombe et qui se relève pour continuer à survivre, nombreux sont ceux qui croient que l'échec et la réussite sont absolus dans la vie."

- Ernest Renan -

Table des matières

Introduction générale

La modélisation des phénomènes n'est pas toujours facile et cela revient à la complexité des modèles, d'ailleurs les méthodes numériques usuelles, ne sont adaptées pour traiter des situations pareilles. Un cadre fréquent de ce type de complexité existe dans de nombreuses disciplines comme la médecine, la physique, le finance, l'économie, ..., où l'étude d'un phénomène fait appel généralement aux modélisations probabilistes qui se terminent souvent par : estimer une quantité, faire des prévisions, tester une hypothèse et même parfois choisir un modèle, comme dans la statistique bayésienne, et tout ça reposent sur les calculs. Mais c'est ainsi que d'autres problèmes commencent, car dire calculs c'est dire erreur de calcul, alors il est important d'avoir des outils permettant d'augmenter la précision ou l'exactitude de ces résultats.

L'augmentation énorme de la puissance des ordinateurs ces dernières années, fait des méthodes de Monte Carlo et MCMC un objet principal face à ces problèmes, pas seulement pour faire des calculs dans des situations complexes, comme par exemple simuler suivant une densité qui n'est pas totalement connue, mais qu'elles proposent également des procédures plus efficaces et plus performantes pour avoir justement une bonne précision, et de rapprocher mieux à la valeur exacte. Pour cette raison alors que les méthodes d'accélération de convergence ou *réduction de variance* prennent leurs place, car grâce à ces méthodes nous pouvons obtenir un résultat dans quelques heures au lieu de quelques jours. Telle situation existe réellement chez quelques disciplines comme la physique nucléaire, la prévision météorologique et l'astronomie, qui utilisent la simulation et la précision comme outils principaux dans leurs recherches. Ces disciplines travaillent d'ailleurs sur des machines de vitesses extrêmes, appelées *super-ordinateurs*, pour avoir justement des résultats d'une façon rapide et précise. Pour ces raisons beaucoup de mathématiciens ont proposé diverses méthodes pour cet objectif bien avant, on cite par exemple: la méthode de stratification, variables antithétiques, variables de contrôle, etc. De même pour nos jours, la recherche ne cesse pas de donner également d'autres alternatives, comme l'approximation Riemannienne développée par Philippe A. et Robert C.P. (2000) et récemment avec Dellaportas Petros Kontoyiannis Ioannis (2012) qui exploitent l'idée des variables de contrôle et l'equation de Poisson pour donner quelques résultats importants.

Chapitre 1

Méthodes de Monte Carlo par Chaînes de Markov (MCMC)

1.1 Chaînes de Markov

Les chaînes de Markov sont les modèles mathématiques les plus simples pour les phénomènes aléatoires évoluant dans le temps. Leurs structures simples permettent d'obtenir beaucoup de résultats importants à propos de leur comportement. En même temps, la classe des ces chaînes est assez riche pour servir dans beaucoup d'applications. Ceci fait pour les chaînes de Markov les premiers et les plus importants exemples des processus aléatoires. En effet, la totalité de l'étude mathématique des processus aléatoires peut être considérée d'une manière ou d'une autre comme une généralisation de la théorie des chaînes de Markov. On peut citer comme références, Benaïm M. et El Karoui N. (2007), Brémaud P. (2009), Caumel Y. (2011) Graham C. (2004), Robert C.P. (1996), Ycart B. (2002),...
Dans cette introduction nous allons donner quelques propriétés importantes pour les chaînes de Markov dans les deux espaces d'état : continu et discret, qui vont nous servir pour mieux comprendre les méthodes MCMC.

1.1.1 Définitions et généralités

Définition 1.1. *Soit $(\Omega, \mathcal{A}, P)$ un espace de probabilité et $\mathcal{X}$ espace des états, une suite de variable aléatoire $(X_n)_{n\in\mathbb{N}}$ à valeurs dans $\mathcal{X}$ est une chaîne de Markov si:*
$\forall n \geq 0$, $\forall x, y, x_{n-2}, \ldots, x_0 \in \mathcal{X}$ *on a:*

$$P(X_n = y \mid X_{n-1} = x, X_{n-2} = x_{n-2}, \ldots, X_0 = x_0) = P(X_n = y \mid X_{n-1} = x).$$

Dans le cas général (cas d'un espace continu) l'idée n'est pas différente.
Une suite de variable aléatoire $(X_n)_{n\in\mathbb{N}}$ est une chaîne de Markov si:

$$P(X_n \in A \mid X_{n-1} = x, X_{n-2} = x_{n-2}, \ldots, X_0 = x_0) = P(X_n \in A \mid X_{n-1} = x_{n-1}).$$

L'idée des chaînes de Markov est très simple, si nous nous intéressons à des évolutions aléatoires, par étapes successives, sur un espace d'états, l'hypothèse fondamentale, appelée *la Propriété de Markov*, sera que l'évolution aléatoire "oublie" son passé et se "régénère" d'instant en instant ne gardant comme information que l'état présent.

Définition 1.2. *La chaîne de Markov est dite* ***homogène*** *(dans le temps), si la probabilité précédente ne dépend pas de n,*

$$P_{x,y} = P(X_n = y \mid X_{n-1} = x), \quad (\forall n \geq 0),$$

appelée probabilité de passage de l'état x à l'état y, en une étape, ou en une opération, ou encore, en une transition.

Définition 1.3. *On appelle noyau de transition, toute fonction P définie sur* $\mathcal{X} \times \mathcal{B}(\mathcal{X})$ *telle que*

i). $\forall x \in \mathcal{X}$, $P(x, \cdot)$ *est une mesure de probabilité;*

ii). $\forall A \in \mathcal{B}(\mathcal{X})$, $P(\cdot, A)$ *est mesurable.*

Dans le cas où $\mathcal{X}$ *est discret, le noyau de transition est une matrice de transition P définie par*

$$P_{x,y} = P(X_n = y \mid X_{n-1} = x), \quad x, y \in \mathcal{X}.$$

Par extension, on appellera également noyau la densité $K(x, x')$ *du noyau* $P(x, \cdot)$*; tel que,*

$$P(X \in A \mid x) = \int_A K(x, x')dx'. \tag{1.1}$$

Exemple 1.1. Transmission d'un message binaire dans une population. Le message reçu par un individu est bien retransmis avec probabilité $1 - p$. L'espace d'état est $E = \{0, 1\}$ (soit {non, oui}). On suppose qu'à l'origine le message est "non", soit $P(X_0 = 0) = 1$. On a

$$P(X_n = 1 \mid X_{n-1} = 1) = P(X_n = 0 \mid X_{n-1} = 0) = 1 - p,$$

$$P(X_n = 1 \mid X_{n-1} = 0) = P(X_n = 0 \mid X_{n-1} = 1) = p$$

donc la matrice de transition s'écrit

$$P = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}$$

1.1.2 Equation de Chapman-Kolmogorov

Une propriété fondamentale des chaînes de Markov est l'équation de Chapman-Kolmogorov. Elle s'agit d'une équation de conservation qui caractérise l'évolution temporelle de la loi du système.
Posant $P^1(x, A) = P(x, A)$, pour $n > 1$, on définit intuitivement

$$P^{n+1}(x, A) = \int_{\mathcal{X}} P^n(y, A)P(x, dy) \quad x \in \mathcal{X},\, A \in \mathcal{B}(\mathcal{X}) \tag{1.2}$$

Le résultat suivant donne des formules de convolution de la forme $P^{m+n} = P^m * P^n$, appelées équations de Chapman-Kolmogorov.

Théorème 1.1. (*Meyn, S.P. and Tweedie R.L. 2009*)
Pour tout (m,n) $\in \mathbb{N}^2$, $x \in \mathcal{X}$, $A \in \mathcal{B}(\mathcal{X})$,

$$P^{n+m}(x, A) = \int_{\mathcal{X}} P^n(y, A)P^m(x, dy), \quad x \in \mathcal{X},\, A \in \mathcal{B}(\mathcal{X}) \tag{1.3}$$

Démonstration 1.1. (*Même référence page 61*)

1.1.2.1 L'operateur P

Il est souvent très utile d'interpréter P comme un opérateur agissant sur les fonctions mesurables et les mesures de façon suivante, l'operateur P transforme une fonction mesurable bornée $f : \mathcal{X} \longrightarrow \mathbb{R}$ en une fonction mesurable bornée
$Pf : \mathcal{X} \longrightarrow \mathbb{R}$ définie par

$$P^n f(x) = \int_{\mathcal{X}} f(y)P^n(x, dy).$$

$$P^n f(x) = \sum_{y \in \mathcal{X}} f(y)P^n_{x,y}, \quad \text{(cas discret)}.$$

et une mesure positive (respectivement une probabilité) μ sur $\mathcal{X}$ en une mesure positive μP définie par

$$\mu P^n(A) = \int_{\mathcal{X}} \mu(dx)P^n(x,A).$$

$$\mu P^n(y) = \sum_{x \in \mathcal{X}} \mu(x)P^n_{x,y}.$$

Théorème 1.2. (*Benaïm Michel et El Karoui Nicole 2007*)
Soit $X = (X_n)$ *une chaîne de Markov sur* $\mathcal{X}$ *de matrice de transition* P. *Alors*

$$P(X_0 = x_0, \ldots, X_n = x) = \mu_0 P_{x_0,x_1} \ldots P_{x_{n-1},x_n} \tag{1.4}$$

où μ_0 désigne la loi de X_0. En particulier,

i. la loi μ_n de X_n vérifie la relation de récurrence (équation de Chapman-Kolmogorov)

$$\mu_{n+1} = \mu_n P = \mu_0 P^{n+1}$$

ii. pour tout $x, y \in \mathcal{X}$

$$P(X_n = y \mid X_0 = x) = P^n_{x,y}$$

iii. pour toute fonction $h : \mathcal{X} \longrightarrow \mathbb{R}$ bornée

$$\mathbb{E}\big[h(X_n) \mid X_0 = x\big] = P^n h(x).$$

Démonstration 1.2. pour (iii)

$$\mathbb{E}\big[h(X_n)\big] = \sum_y h(y)\mu_n(y) = \sum_y h(y)(\mu P^n)(y)$$

Pour $\mu_0 = \delta_x$, avec δ_x mesure de Dirac, on en déduit que

$$P(X_n = y \mid X_0 = x) = \sum_y h(y) P^n_{x,y} = P^n h(x)$$

Définition 1.4. *Soit $A \in \mathcal{B}(\mathcal{X})$. On note*

$$\tau_{\mathcal{A}} = \inf\{n \geq 1;\ X_n \in A\}$$

le temps d'arrêt en A ou le premier instant où la chaîne rentre dans A avec, par convention, $\tau_{\mathcal{A}} = +\infty$ si $X_n \notin A$ pour tout n.

On définit également

• Le nombre de passages de (X_n) en A

$$\eta_A = \sum_{n=1}^{\infty} \mathbb{1}_A(X_n).$$

• Le nombre moyen de passages en A

$$U(x, A) = \mathbb{E}_x[\eta_A].$$

• La probabilité de retour en A en un nombre fini d'étapes

$$L(x, A) = P_x(\eta_A < \infty).$$

1.1.3 Irréductibilité

La propriété d'irréductibilité est importante dans le contexte des algorithmes de Monte Carlo par chaînes de Markov, elle est une première mesure de la robustesse de la chaîne de Markov face aux conditions initiales, μ_0.

Définition 1.5. *On dit que l'état y est **accessible** à partir de x (noté $x \to y$) s'il existe $n \geq 0$ tel que $P_x(X_n = y) > 0$. On dit que les états x et y **communiquent** (noté $\leftrightarrow$) si $x \to y$ et $y \to x$.*

Définition 1.6. *La chaîne est **irréductible** si tous les états communiquent entre eux, c'est-à-dire si*

$$L(x, y) > 0, \quad \forall x, y \in \mathcal{X}$$

Dans le cas général (c'est-à-dire non discret) $L(x, y)$ est nul à de rares exceptions, alors et il faut introduire une mesure auxiliaire φ sur $\mathcal{B}(\mathcal{X})$ pour définir proprement cette notion.

Définition 1.7. *Etant donnée une mesure φ , la chaîne (X_n) est dite φ-irréductible si, pour tout $A \in \mathcal{B}(\mathcal{X})$, $\varphi(A) > 0$ entraîne*

$$L(x, A) > 0, \quad \forall x \in \mathcal{X}$$

Proposition 1.1. (*Robert C.P. 1996*)
La chaîne (X_n) est φ-irréductible si, et seulement si, pour tout $x \in \mathcal{X}$ et tout $A \in \mathcal{B}(\mathcal{X})$ tel que $\varphi(A) > 0$, on a l'une des propriétés suivantes :

(i) il existe $n \in \mathbb{N}$ tel que $P^n(x, A) > 0$;

(ii) $U(x, A) > 0$

Exemple 1.2. Soit la chaîne suivante $X_{n+1} = \theta X_n + \epsilon_{n+1}$ et ϵ_n une variable normale, la chaîne est bien irréductible, en effet, $P(x, A) > 0$ pour tout $x \in \mathbb{R}$ et tout A tel que $\lambda(A) > 0$, (λ *étant la mesure de Lebesgue*). Par contre, si ϵ_n est uniforme sur $[-1, 1]$ et $|\theta| > 1$, la chaîne n'est plus irréductible. Par exemple, si $\theta > 1$, il vient que

$$X_{n+1} - X_n \geq (\theta - 1)X_n - 1 \geq 0$$

pour $X_n > 1/(\theta - 1)$. La chaîne croît donc de manière monotone.

Illustration graphique

Dans cet exemple, on donne deux simulation pour la chaîne (X_n).
La première: on prend avec $\theta = 0.5$ et ϵ_n suit une loi $\mathcal{N}(0, 1)$ et un ensemble $A = [0, 1]$

tel que $\lambda(A) = 1 > 0$.
La deuxième: on prend avec $\theta = 1.2$ et ϵ_n suit une loi uniforme sur $[-1, 1]$ et un ensemble $B = [40,20]$ tel que $\lambda(B) = 20 > 0$.

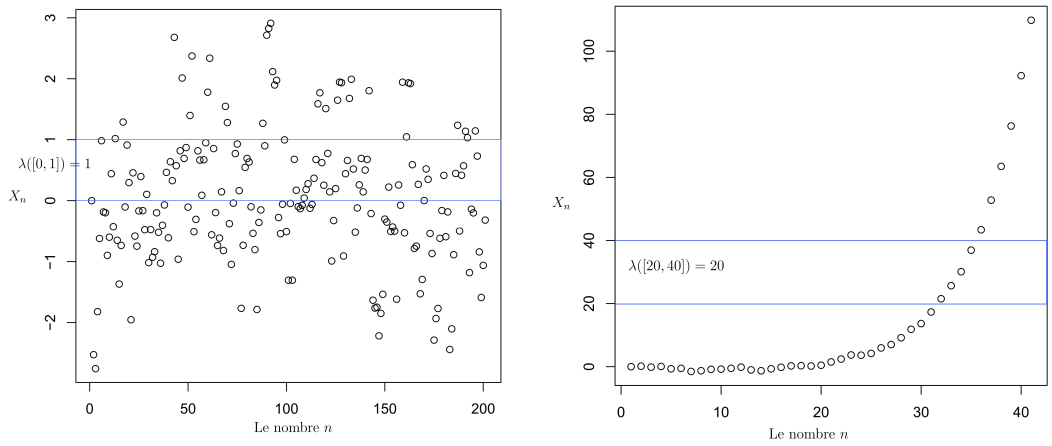

FIG 1.1. Evolution de (X_n) pour ϵ_n une loi $\mathcal{N}(0, 1)$ à gauche et loi uniforme sur $[-1, 1]$ à droite.

On remarque sur la figure à gauche que la chaîne revient à la bande plusieurs fois, contrairement à la figure à droite où la chaîne passe mais sans retour.

1.1.4 Transience, récurrence et période

Définition 1.8. *L'état $x \in \mathcal{X}$ est dit* ***récurrent*** *si $P_x(\eta_x < \infty) = 1$, récurrent positif si de plus $\mathbb{E}_x[\eta_x] < \infty$, et récurrent nul si $\mathbb{E}_x[\eta_x] = \infty$, dans le cas contraire i.e. $P_x(\eta_x < \infty) < 1$, l'état x est dit* ***transitoire****.*

Le fait qu'un état soit récurrent signifie que la chaîne revient vers cet état presque surement, et donc qu'elle y revient infiniment souvent. Le fait qu'un état soit transient signifie que la chaîne a une probabilité positive de ne jamais retourner dans cet état.

Définition 1.9. *Dans le cas général une chaîne (X_n) est récurrente s'il existe une mesure φ telle que (X_n) soit φ-irréductible et si $\mathbb{E}_x[\eta_A] = \infty$ pour tout $x \in \mathcal{X}$ et pour tout $A \in \mathcal{B}(\mathcal{X})$ tel que $\varphi(A) > 0$. Elle est transiente si elle est φ-irréductible et si $\mathcal{X}$ est transient.*

1.1.5.1 Récurrence au sens de Harris

Il est possible de renforcer la stabilité de la chaîne (X_n) en imposant non seulement un nombre moyen infini de passages dans tout ensemble mais aussi la garantie d'un nombre infini de passages pour presque toute trajectoire, suivant la notion introduite

par Harris (1956).

Définition 1.10. *Un ensemble A est récurrent au sens de Harris si*

$$Q(x, A) := P_x(\eta_A = \infty) = 1, \qquad \forall x \in A.$$

La chaîne (X_n) est récurrente au sens de Harris s'il existe une mesure φ telle que (X_n) soit φ-irréductible et si tout ensemble A tel que $\varphi(A) > 0$ est récurrent au sens de Harris.

Proposition 1.2. (*Robert C.P. 1996*)
Si, pour tout $A \in \mathcal{B}(\mathcal{X})$, on a $L(x, A) = 1$ pour tout $x \in A$, alors

$$Q(x, A) = L(x, A), \quad \forall x \in \mathcal{X},$$

et (X_n) est récurrente au sens de Harris.

Démonstration 1.3. (*Dans la même référence page 107*)

Définition 1.11. *La période d'un état $x \in \mathcal{X}$ est le nombre*

$$d_x = p.g.c.d. \{n \geq 0 : \ P_x(X_n = x)\}$$

Si $d_x = 1$, on dit que l'état x est apériodique. Si tout $x \in \mathcal{X}$ est apériodique, on dit que la chaîne est apériodique.

Dans le cas général, c'est-à-dire quand le noyau de transition admet une densité $f(\cdot|x_n)$ par rapport à la mesure de Lebesgue, une condition suffisante d'apériodicité est que $f(\cdot|x_n)$ soit positive dans un voisinage de x_n puisqu'on peut alors rester un nombre arbitraire d'instants dans ce voisinage avant de visiter un ensemble A quelconque.
Dans le contexte des algorithmes de Monte Carlo par chaînes de Markov, nous allons voir que ces algorithmes conduisent bien à des chaînes apériodiques.

Définition 1.12. *Une chaîne de Markov (X_n) est dite récurrente irréductible si elle est irréductible et si tous les états sont récurrents.*

Proposition 1.3. *[4] Toute chaîne de Markov irréductible sur un espace **fini** $\mathcal{X}$ est récurrente irréductible.*

Définition 1.13. *Une chaîne de Markov homogène irréductible récurrente positive et apériodique est dite **ergodique**.*

1.1.5 Mesures invariantes

Les chaînes de Markov produites par les méthodes de Monte Carlo par chaînes de Markov possèdent par construction cette propriété de plus grande stabilité.

Définition 1.14. *Une distribution de probabilité π sur $\mathcal{X}$ est dite* ***invariante*** *ou* ***stationnaire*** *si elle satisfait*

$$\pi = \pi P. \tag{1.5}$$

Dans le cas continu

$$\pi(A) = \int_{\mathcal{X}} P(x, A)\pi(dx), \qquad \forall A \in \mathcal{B}(\mathcal{X}). \tag{1.6}$$

Lorsqu'il existe une mesure de probabilité invariante pour une chaîne φ-irréductible, la chaîne est dite ***positive****.*

On parlera également de loi stationnaire dans le cas où π est une mesure de probabilité car $X_0 \sim \pi$ entraîne que $X_n \sim \pi$ pour tout n, donc que la chaîne est stationnaire.

1.1.6 La réversibilité

La notion de retournement dans temps où réversibilité est très productive en théorie des chaînes de Markov.

Définition 1.15. *Soit π une probabilité sur $\mathcal{X}$. La matrice de transition P (respectivement la chaîne (X_n)) est dite* ***réversible*** *par apport à π si*

$$\pi(x)P_{x,y} = \pi(y)P_{y,x} \tag{1.7}$$

pout tout $x, y \in \mathcal{X}$.
Dans le cas où $\mathcal{X}$ est continu

$$\pi(dx)P(x, dy) = \pi(dy)P(y, dx) \tag{1.8}$$

Proposition 1.4. (*Roberts G. O. Rosenthal J. S. 2004*)
Si P est réversible par apport à π alors π est une probabilité invariante.

Démonstration 1.4. Soit μ une probabilité invariante. Alors μ est caractérisée par le système d'équations

$$\mu(y) = \sum_x \mu(x)P(x, y) = \mu(x)\left(1 - \sum_{x:x\neq y} P(y, x)\right) + \sum_{x:x\neq y} \mu(x)P(x, y).$$

D'où

$$\mu(y)\sum_{x:x\neq y}P(y,x)=\sum_{x:x\neq y}\mu(x)P(x,y)$$

Sous cette forme, il est évident que π est solution.
Pour (1.8):

$$\begin{aligned}\int_{x\in\mathcal{X}}\pi(dx)P(x,dy)&=\int_{x\in\mathcal{X}}\pi(dy)P(y,dx)\\&=\pi(dy)\int_{x\in\mathcal{X}}P(y,dx)\\&=\pi(dy).\end{aligned}$$

1.1.7 Théorème ergodique

Théorème 1.3. *[4] (Théorème ergodique)*
Supposons P irréductible de mesure invariante π. Alors, pour toute mesure initiale μ, et tout $x\in\mathcal{X}$

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\mathbb{1}_{\{X_i=x\}}=\pi(x),\quad p.s. \tag{1.9}$$

Sous les mêmes hypothèses, pour toute fonction $h:\mathcal{X}\longrightarrow\mathbb{R}$,

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}h(X_i)=\int h\,d\pi,\quad p.s. \tag{1.10}$$

1.1.8 Equation de Poisson

Beaucoup des développements sont basés sur l'identité suivante, connue sous le nom *d'équation de Poisson* ou *fonction de Green*, qui joue un rôle central dans les méthodes MCMC.
Soient $f:\mathcal{X}\longrightarrow\mathbb{R}$ et P un noyau de transition sur $\big(\mathcal{X},\mathcal{B}(\mathcal{X})\big)$;

$$\hat{f}-P\hat{f}=f-\pi(f) \tag{1.11}$$

La fonction f s'appelle la *fonction forçante (forcing function)*. Dans la plupart des cas la fonction forçante est donnée, et $\hat{f}$ s'appelle la solution de l'équation de Poisson.

1.2 Méthode de Monte Carlo

L'idée principale de la méthode de Monte Carlo est de rapprocher une valeur prévue $\mathbb{E}(X)$ par une moyenne arithmétique des résultats d'un grand nombre d'expériences indépendantes et de même distribution. La base de cette méthode est l'un des résultats

les plus célébrés de la théorie des probabilités, la loi forte de grands nombres. Car les valeurs prévues jouent un rôle central dans divers domaines d'applications de la modélisation probabiliste.

Historiquement, la méthode remonte au Comte de Buffon qui, en 1777, a décrit une méthode de calcul de π basée sur la réalisation d'expériences répétées. Mais la vraie naissance de la méthode de Monte Carlo est liée à l'apparition des premiers ordinateurs et à leurs utilisations dans le cadre des projets secrets du département de la défense des Etats Unis dans les années $40-45$. Les premiers articles décrivant ce type de méthodes datent de la fin des années 40 et du début des années 50. Ces méthodes numériques sont maintenant de plus en plus utilisées, à cause de leur facilité de programmation, et de la possibilité de réaliser en un temps raisonnable un nombre gigantesque de tirages aléatoires sur les ordinateurs modernes.
Les domaines d'application sont nombreux. Elles sont utilisées notamment en statistique, Probabilité, optimisation, l'analyse, systèmes d'attente (comme dans des supermarchés ou dans de grandes usines), l'analyse de la fiabilité des systèmes techniques, la conception des réseaux de télécommunication, l'évaluation des risques des investissements etc.

1.2.1 Description de la méthode

Pour donner une première idée sur la méthode de Monte Carlo, on considérer un problème d'intégration numérique suivant:

$$\mathcal{I} = \int_a^b h(x)dx \tag{1.12}$$

Où la fonction h n'est pas intégrable[1]. Il existe alors de nombreuses méthodes permettant d'approximer cette intégrale, comme par exemple, la méthode de trapèze, Gauss, Simpson..., qui sont des méthodes analystes. Mais celle de Monte Carlo est tout à fait différente, car elle fait appel à la théorie des probabilités pour approcher cette intégrale avec une précision satisfaisante dans beaucoup de cas.
Soit alors une variable aléatoire $(v.a)$ X, uniforme sur $[a,b]$, de densité

$$f(x) = \frac{1}{b-a}\mathbb{1}_{(a,b)}(x)$$

Si on réécrit (1.12) en posant

$$\varphi(x) = \frac{h(x)}{f(x)} = (b-a)h(x)\mathbb{1}_{(a,b)}(x),$$

on aura alors

$$\mathcal{I} = \int_a^b \frac{h(x)}{f(x)} f(x)dx = \int_a^b \varphi(x)f(x)dx = \mathbb{E}_f[\varphi(x)],$$

1. Valable même dans cas où h est intégrable.

d'autre part, et d'après la loi des grands nombres, pour n assez grand on a

$$\mathbb{E}_f[\varphi(x)] = \frac{1}{n}\sum_{i=1}^{n}\varphi(X_i)$$

Finalement l'approximation de Monte Carlo pour cette intégrale sera

$$\int_a^b h(x)dx \simeq \frac{b-a}{n}\sum_{i=1}^{n} h(X_i) = \bar{h}_n \tag{1.13}$$

D'où, le principe de la méthode de Monte-Carlo pour approcher $\mathcal{I}$ est: on génére (par ordinateur) un échantillon $(X_1,\ldots,X_n)$ suivant la densité uniforme et de l'approximer par la moyenne empirique.

1.2.2 Application

L'approximation de la méthode de Monte Carlo est utilisée généralement chaque fois que les fonctions à intégrer sont très compliquées, où les calculs avec les autres méthodes sont difficiles et/ou longs .

Exemple 1.3. On désire calculer cette intégrale compliquée

$$\delta = \int_{0.2}^{2} x\sqrt{\exp\left(\frac{\cos\log(x)}{x^2}\right)}dx \tag{1.14}$$

Il est clair que la tentative de résoudre ce problème avec les méthodes analystes sera une perte de temps. Mais en utilisant la méthode de Monte Carlo la tâche devient très simple.
On génère alors un échantillon $(X_1,\ldots,X_n)$ de taille $n = 10^5$ avec X uniforme sur $[0.2, 2]$ et on approche δ par,

$$\bar{h}_n = \frac{1.8}{10^5}\sum_{i=1}^{10^5} X_i\sqrt{\exp\left(\frac{\cos\log(X_i)}{X_i^2}\right)}$$

Le programme en `R` est comme suit:

Programme 1.1.

```
n=100000; a=0.2; b=2; s=0;
h=function(x) {x*sqrt(exp(cos(log(x))/(x^2)))}
plot(h,from=0,to=3) # pour voir le graphe
for (i in 1:n){
u=runif(1,a,b)
s=s+h(u)}
s*(b-a)/n
```

Ou simplement

Programme 1.2.

```
n=100000; a=0.2; b=2;
h=function(x) {x*sqrt(exp(cos(log(x))/(x^2)))}
(b-a)*mean(h(runif(n,a,b)))
```

La valeur donnée est : $\bar{h}_n = 3.405161$.

Remarque 1.
On préfère le programme (1.2) à cause de sa rapidité de fournir les résultats.

1.2.3 Contrôle de convergence avec le Théorème de la Limite Centrale

Quand $h^2(X)$ a une variance finie sous f, la vitesse de convergence de $\bar{h}_n$ peut être évaluée car la convergence se produit à la vitesse $O(\sqrt{n})$ et la variance asymptotique de l'approximation est

$$\mathbb{V}\text{ar}(\bar{h}_n) = \frac{1}{n}\int \Big(h(x) - \mathbb{E}_f\big[h(X)\big]\Big)^2 f(x)dx, \tag{1.15}$$

qu'on peut l'estimer également par,

$$\upsilon_h = \frac{1}{n^2}\sum_{i=1}^{n}\Big[h(x_i) - \bar{h}_n\Big]^2$$

D'après le théorème de la Limite Centrale, pour n assez grand on a

$$\frac{\bar{h}_n - \mathbb{E}_f[h(X)]}{\sqrt{\upsilon_h}} \xrightarrow{L} \mathcal{N}(0,1), \tag{1.16}$$

cela conduit bien à un test de convergence et aux bornes de confiance sur l'approximation de $\mathbb{E}_f\big[h(X)\big]$.

Exemple 1.4. suite à l'exemple (1.3).
La figure (1.2) à droite montre comment la convergence se produit, ainsi que les bornes dérivées des erreurs standard estimées en fonction du nombre n de simulations.

Remarque 2. On peut également augmenter le nombre de simulation pour avoir une bonne (même excellente) approximation comme la montre la figure (1.2) à gauche, avec la bande bleue plus serrée

L'implémentation en R est comme suit :

Programme 1.3.

```
xh=(b-a)*(h(runif(n,a,b)))
esth=cumsum(xh)/(1:n)
bornes=sqrt(cumsum((xh-esth)^2))/(1:n)
```

```
plot(esth,type="l", xlab="Moyenne et variation",lwd=2)
lines(esth+2*bornes,col="blue", lwd=1)
lines(esth-2*bornes,col="blue", lwd=1)
```

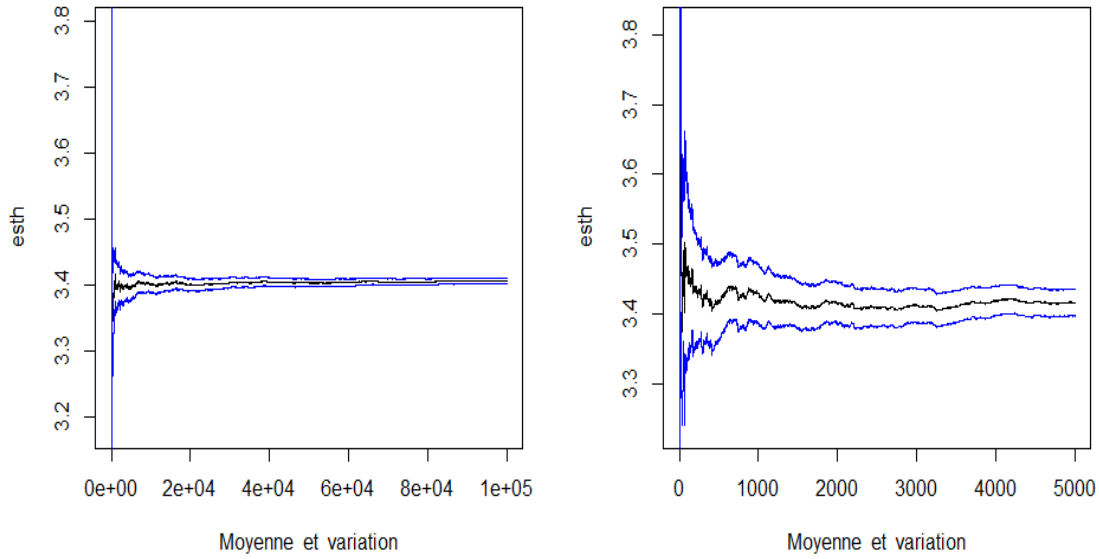

FIG 1.2. Approximation de l'intégrale de la fonction (1.14) avec n différent

1.2.4 Inconvénients

a)- Les bornes d'intégration

Le fait que cette méthode dépende des bornes d'intégration de la quantité à estimer pose deux problèmes importants:

1. Il faut connaître les bornes a et b d'une façon exacte, car ce sont les paramètres même de la loi uniforme.
2. La distance $|b-a|$ entre ces bornes est une question cruciale. En effet, chaque fois que cette distance augmente, la variance augmente avec elle en diminuant ainsi la précision..

Exemple 1.5. Supposons qu'on veut estimer l'intégrale suivante

$$\mathcal{I} = \int_a^b \lambda e^{-\lambda x} dx \quad \text{avec} \quad \lambda = 15, \tag{1.17}$$

par l'estimateur de Monte Carlo

$$\bar{h}_{(a,b)} = \frac{b-a}{10^5} \sum_{i=1}^{10^5} \lambda exp(-\lambda X_i)$$

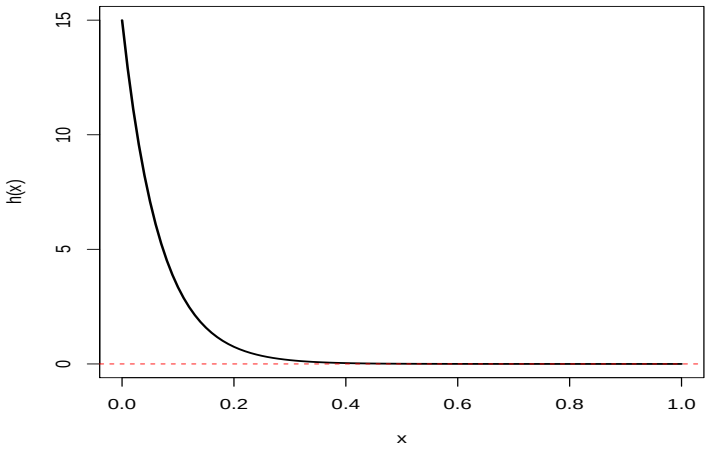

FIG 1.3. Le graphe de la fonction à intégrer

D'après la figure [1.3], il est clair que la surface est presque la même entre les valeurs $a = 0$ jusqu'à $b = 0.4$ et de $a = 0$ à $b = 10$,
C'est-à-dire,

$$\int_0^{10} \lambda e^{-\lambda x} dx \simeq \int_0^{0.4} \lambda e^{-\lambda x} dx \quad \text{implique} \quad \bar{h}_{(0,10)} \simeq \bar{h}_{(0,0.4)}.$$

Mais la différence entre les deux estimations données avec la méthode de Monte Carlo est flagrante. Sur la figure [1.4] on voit bien que la variance des deux estimateurs est très différente.

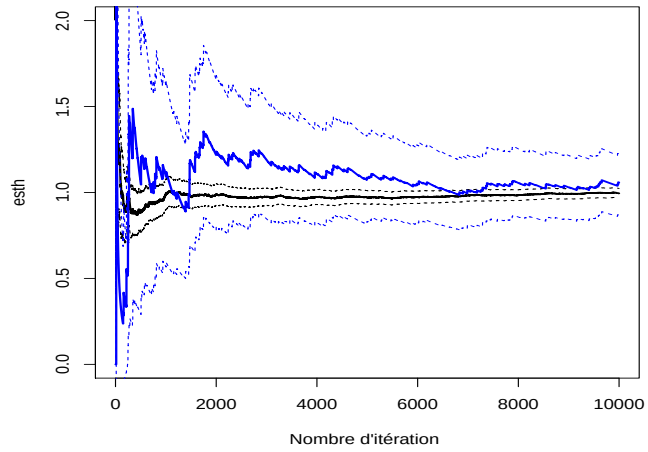

FIG 1.4. Approximation de l'intégrale (1.17) en fonction des itérations

Le code R correspondant est :

Programme 1.4.

```
n=10000; a=0; b=.4;
h=function(x,a=15) {a*exp(-a*x)}
curve(h,lwd=2)
abline(a=0,b=0,col=2,lty=2)
x=h(runif(n,a,b))
x=(b-a)*(h(runif(n,a,b)))
esth=cumsum(x)/(1:n)
bornes=sqrt(cumsum((x-esth)^2))/(1:n)
plot(esth,type="l",xlab="Nombre d'itération",lwd=2,ylim=c(0,2))
lines(esth+2*bornes,lty=2)
lines(esth-2*bornes,lty=2)
a=0; b=10;
y=h(runif(n,a,b))
y=(b-a)*(h(runif(n,a,b)))
esthy=cumsum(y)/(1:n)
bornes=sqrt(cumsum((y-esthy)^2))/(1:n)
lines(esthy,type="l",col=4,lwd=2)
lines(esthy+2*bornes,lty=2,col=4)
lines(esthy-2*bornes,lty=2,col=4)
var(y)/var(x) # le rapport des deux variances
```

Le rapport des deux variances donné, est : $\frac{\bar{h}_{(0,0.4)}}{\bar{h}_{(0,10)}} = 43.44$, c'est-à-dire que l'estimateur $\bar{h}_{(0,0.4)}$ est 43 fois plus précis en comparant avec l'estimateur $\bar{h}_{(0,10)}$.

b)- La connaissance totale de la fonction à intégrer

Un autre cas où la méthode de Monte Carlo ne peut pas s'appliquer correctement, est quand la fonction est définie à une constante près (ce cas est toujours rencontré en statistique Bayésienne),

$$\mathcal{I} = \int_a^b h(x)dx = \int_a^b \lambda f(x)dx \tag{1.18}$$

D'après la formule (1.13) nous allons aboutir à,

$$\lambda = \frac{\bar{f}_n}{\mathcal{I}},$$

qui veut dire: pour connaître la constante λ il faut connaître $\mathcal{I}$, mais ça c'est l'objectif voulu !

1.3 Méthodes MCMC(1): Algorithmes de Hastings-Metropolis

Les méthodes de simulation par chaîne de Markov se sont considérablement développées ces dernières années. Elles sont utilisées dans des situations aussi variées comme l'optimisation combinatoire, le traitement d'image, le finance, ou encore l'expérimentation numérique en physique statistique...

1.3.1 Motivation

Jusqu'ici nous avons généré des variables indépendantes et identiquement distribuées (iid), directement à partir de la densité d'intérêt f comme le cas des méthodes de Monte Carlo. Les méthodes MCMC présentées ci-dessous génèrent par contre des variables corrélées en utilisant les chaînes de Markov. La raison de ce changement est que les chaînes de Markov ont des propriétés de convergence différentes qui peuvent être exploitées pour fournir des propositions plus simples, D'une part, cette procédure nous permet de s'en sortir dans des situations où l'information sur la loi cible f est peu (cas où $f \propto g$) ou même médiocre, où les méthodes "*classiques*" ne peuvent pas résister. D'autre part, il est souvent impossible de construire rapidement un algorithme efficace de simulation suivant une loi donnée.

1.3.2 Le Rôle des chaînes de Markov

On rappelle qu'une chaîne de Markov $(X_t)_{t\in\mathbb{N}}$ est une suite de variables aléatoires dépendantes

$$X_0, X_1, X_2, \ldots, X_t, \ldots$$

telles que X_t sachant les variables passées ne dépend que de X_{t-1}. On appelle cette probabilité conditionnelle un *noyau de transition* ou un *noyau markovien* K ; c'est-à-dire,

$$X_{t+1} \mid X_0, X_1, \ldots, X_t \sim K(X_t, X_{t+1})$$

Dans la plupart des cas, les chaînes de Markov qui interviennent dans le cadre des algorithmes de Monte Carlo par chaînes de Markov satisfont une propriété de stabilité très forte. En effet, il existe une loi de probabilité par construction pour ces chaînes, c'est-à-dire une loi de probabilité f telle que, si $X_t \sim f$ alors $X_{t+1} \sim f$. Ainsi, formellement, le noyau et la loi stationnaire satisfont l'équation

$$\int_{\mathcal{X}} K(x, y) f(x) dx = f(y) \tag{1.19}$$

pour avoir la stationnarité, le critère *d'irréductibilité* pour la chaîne doit être vérifié, en effet, dans ce cas la chaîne peut visiter tout l'espace d'état, à savoir, quelle que soit la valeur initiale X_0, la probabilité que la suite (X_t) atteigne n'importe quelle région

de l'espace d'état est non nulle. (Une condition suffisante est que $K(x,\cdot\,) > 0$). L'existence d'une loi stationnaire a des conséquences importantes sur le comportement de la chaîne (X_t). Notamment, la plupart des chaînes qui interviennent dans des algorithmes MCMC sont *récurrentes*, c'est-à-dire qu'elles visitent dans leur histoire tout ensemble non négligeable un nombre infini de fois. Dans le cas de chaînes récurrentes, la loi stationnaire est aussi une loi limite dans le sens où la loi limite de X_t est f pour presque toute valeur initiale X_0. Cette propriété, aussi appelée *ergodicité*, a des conséquences majeures du point de vue des simulations. En effet, si un noyau donné K induit une chaîne de Markov ergodique de loi stationnaire f, alors générer une chaîne à partir de ce noyau K induira finalement des simulations selon f. De plus, pour des fonctions intégrables h, la moyenne standard vérifie

$$\frac{1}{T}\sum_{t=1}^{T} h(X_t) \longrightarrow \mathbb{E}_f\big[h(X)\big] \tag{1.20}$$

On l'appelle le *Théorème Ergodique*.

1.3.3 Algorithmes de Metropolis-Hastings (M-H)

Un algorithme quasi universel satisfaisant, développé par Metropolis et al. (1953)[2], au départ pour la physique particulaire (et la bombe H...), et généralisé par Hastings (1970) dans un cadre plus statistique (et plus pacifique). En réalité, il s'applique à une grande variété de problèmes, il repose sur l'utilisation d'un noyau Markovien K de loi stationnaire f, puis on génère une chaîne de Markov (X_t) en utilisant ce noyau, de telle sorte que la loi limite de (X_t) soit f.

Algorithme 1.1. *(Metropolis-Hastings)*
Étant donné x_t;

1. *Générer $y_t \sim q(y/x_t)$;*
2. *prendre*

$$x_{t+1} = \begin{cases} y_t & \text{avec probabilité } \rho(x_t,y_t) \\ x_t & \text{avec probabilité } 1-\rho(x_t,y_t) \end{cases}$$

avec

$$\rho(x_t,y_t) = \min\left\{\frac{f(y_t)}{f(x_t)}\frac{q(x_t/y_t)}{q(y_t/x_t)},1\right\}$$

On appelle q la loi de *proposition* (ou loi *instrumentale* ou loi *candidate*) et la probabilité $\rho(x,y)$ la probabilité *d'acceptation* de Metropolis-Hastings.

Remarque 3.
Un avantage apporté par l'algorithme M-H est :

2. Disponible sur: https://ssl.cs.dartmouth.edu/ gevorg/89/13W/Metropolis_MC.pdf

Si on cherche à générer des lois ayant une fonction de densité connue à une constante prés (*i.e.* $f = \lambda h$) avec λ inconnue, alors il suffit de remarquer que

$$\frac{f(y_t)}{f(x_t)}\frac{q(x_t/y_t)}{q(y_t/x_t)} = \frac{\lambda h(y_t)}{\lambda h(x_t)}\frac{q(x_t/y_t)}{q(y_t/x_t)} = \frac{h(y_t)}{h(x_t)}\frac{q(x_t/y_t)}{q(y_t/x_t)}$$

Donc la constante n'a pas d'influence sur l'échantillon généré, et le problème est aisément dépassé. Par contre, la méthode de Monte Carlo déjà présentée ne peut pas s'échapper à ce genre de problème.

Théorème 1.4. (*Robert C.P. 1996*)
Pour toute loi conditionnelle q, f est une loi stationnaire de la chaîne (X_t) produite par algorithme (1.1).

Démonstration 1.5. Le noyau de transition associé à l'algorithme (1.1) sécrit

$$K(x,x') = \rho(x,x')q(x'|x) + (1-\rho(x,x'))\delta_x(x')$$

où δ_x désigne la masse de Dirac en x. On a donc, pour tout ensemble mesurable A,

$$\begin{aligned}
\int K(x,A)f(x)dx &= \int\int \mathbb{1}_A(x')\rho(x,x')q(x'|x)f(x)dxdx' \\
&+ \int\int (1-\rho(x,x'))q(x'|x)dx'\mathbb{1}_A(x)f(x)dxdx' \\
&= \int\int_D \mathbb{1}_A(x')\frac{f(x')}{f(x)}\frac{q(x|x')}{q(x'|x)}q(x'|x)f(x)dxdx' \\
&+ \int\int_{D^c} \mathbb{1}_A(x')q(x'|x)f(x)dxdx' \\
&+ \int\int_D \mathbb{1}_A(x)\left(1-\frac{f(x')}{f(x)}\frac{q(x|x')}{q(x'|x)}\right)q(x'|x)f(x)dxdx'
\end{aligned}$$

pour $D = \{(x',x); f(x')q(x|x') \leq f(x)q(x'|x)\}$. Ainsi

$$\begin{aligned}
\int K(x,A)f(x)dx &= \int\int \mathbb{1}_A(x')f(x')q(x|x')dxdx' \\
&+ \int\int_D \mathbb{1}_A(x)f(x)q(x'|x)dxdx' \\
&+ \int\int_{D^c} \mathbb{1}_A(x')f(x)q(x'|x)dxdx' \\
&- \int\int_D \mathbb{1}_A(x)f(x)q(x|x')dxdx' \\
&= \int\int \mathbb{1}_A(x')f(x')q(x|x')dxdx' \\
&= \int_A f(x')dx',
\end{aligned}$$

en opérant le changement de variables de (x,x) à (x,x') sur les seconde et quatrième intégrales de la première égalité, puisqu'alors D^c est transformé en D et réciproquement.

□

La stationnarité de f est donc assurée quelle que soit la loi conditionnelle q, ce qui donne une mesure de l'universalité des algorithmes de Hastings-Metropolis.

1.3.4 Algorithme de Metropolis-Hastings indépendant

D'après le théorème (1.4) on peut utiliser également une loi de proposition q qui ne dépend pas de l'état présent de la chaîne, (c'est-à-dire $q(y|x) = g(y)$), donc on obtient un cas particulier de l'algorithme de départ.

Algorithme 1.2.
Étant donné x_t;

1. *Générer $y_t \sim g(y)$;*
2. *prendre*

$$x_{t+1} = \begin{cases} y_t & \text{avec probabilité} \quad \min\left\{\frac{f(y_t)}{f(x_t)}\frac{q(x_t)}{q(y_t)},1\right\} \\ x_t & \text{Sinon} \end{cases}$$

Exemple 1.6. (Loi de Gamma à partir de loi normale)
Pour générer une variable aléatoire Gamma $\Gamma(2,4)$ de densité,

$$f(x) = \frac{4^2}{\Gamma(2)} x e^{-4x} \mathbb{1}_{[0,\infty[}(x),$$

Nous utilisons une loi de proposition $\mathcal{N}(0,1)$ pour l'algorithme de Metropolis-Hastings.

Le code R correspondant :

Programme 1.5.

```
N=100000; a=2; b=4; mu=0; bn=40;
x=numeric(1); x[1]=.5;
f=function(x) {((a^b)/gamma(a))*(x^(a-1))*exp(-b*x)}
g=function(x) {(1/sqrt(pi*2))*exp(-(1/2)*(x-mu)^2)}
for(i in 2:N){
y=rnorm(1,mu,1)
if(runif(1)<min((f(y)*g(x[i-1]))/(f(x[i-1])*g(y)),1))
x[i]=y
else {x[i]=x[i-1]}}
hist(x,breaks=bn,prob=TRUE,main="Echantillon généré par l'algorithme M-H")
```

```
curve(f,add=TRUE,col="blue",lty=1,lwd=2)
```

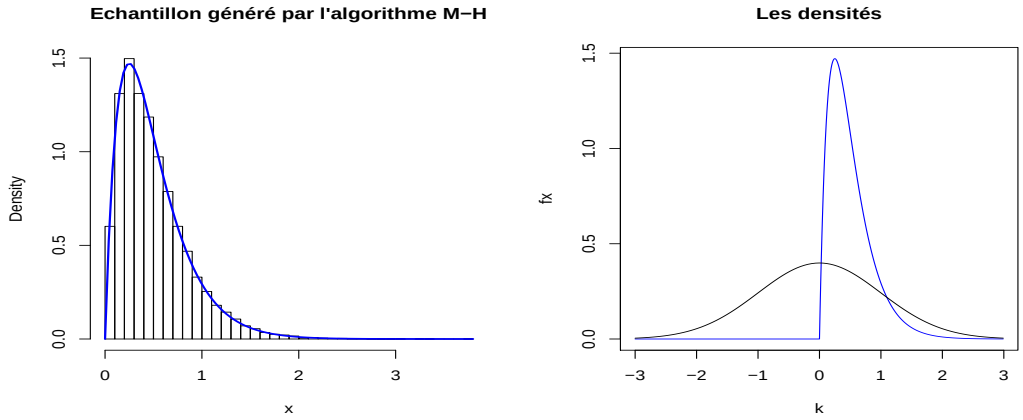

FIG 1.5. Histogramme de l'échantillon avec la densité Gamma superposée et les deux lois, cible et instrumentale

On remarque sur histogramme la ressemblance entre l'échantillon simulé par l'algorithme M-H et loi cible tracée en bleu, et à droite la densité cible et la densité de proposition en noir.

Remarque 4.
L'échantillon obtenu par l'algorithme M-H n'est pas iid. Même si les Y_t soient générés de manière indépendante, parce que la probabilité d'acceptation de Y_t dépend de X_t. On peut le voir à l'aide de la fonction acf de R disponible dans le package *tseries*.

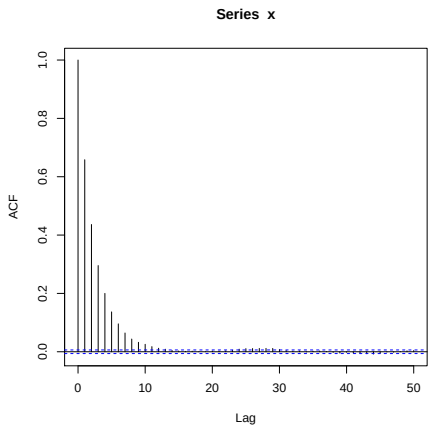

FIG 1.6. Autocovariances pour un échantillon généré par algorithme M-H

Inconvénient

Cette méthode demande généralement une connaissance plus approfondie de la distribution cible pour une convergence optimale. Par exemple, il est préférable que les régions de forte densité de la distribution instrumentale coïncident avec les régions de forte densité de la distribution cible. D'un autre côté, il est souhaitable que l'épaisseur des queues des deux densités soit relativement similaire afin de ne pas sous-estimer certaines régions.
Si on remplace dans l'exemple précédent la loi de proposition $\mathcal{N}(0,1)$ par $\mathcal{N}(0,5)$ on aura des problèmes pour la stationnarité.

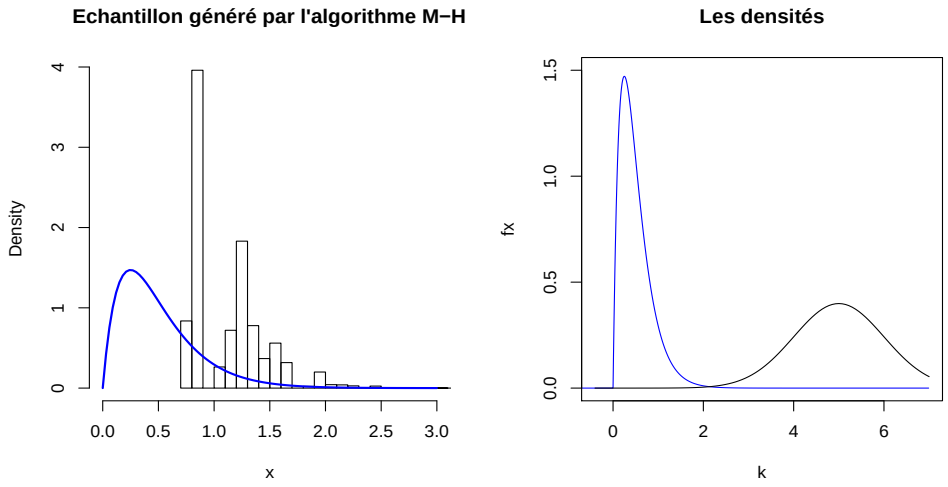

FIG 1.7. Histogramme avec la densité Gamma superposée et la loi cible avec la loi instrumentale

La figure [1.7] à gauche montre bien ce problème. En effet, l'échantillon généré avec l'algorithme M-H ne coïncide pas avec la densité cible représentée en bleu. De l'autre côté, les deux densités : cible en bleu et instrumentale en noir, avec les régions de forte densité très éloignées.

1.3.5 Metropolis-Hastings par marche aléatoire

Une autre approche naturelle pour la construction pratique d'un algorithme de Metropolis-Hastings est de prendre en compte la valeur précédemment simulée pour générer la valeur suivante, c'est-à-dire, à explorer localement le voisinage de la valeur présente de la chaîne de Markov.
On génère alors Y_t de la manière suivante

$$Y_t = X_t + \epsilon_t$$

où ϵ_t est une perturbation aléatoire de loi g indépendante de X_t , par exemple une loi uniforme ou normale, signifiant que $Y_t \sim U(X_t - \gamma,\, X_t + \gamma)$ ou $Y_t \sim \mathcal{N}(X_t,\, \tau^2)$. Dans ce cas, la densité de proposition $q(y|x)$ sera de la forme $g(y - x)$ et la chaîne de Markov associée à q est une marche aléatoire où la densité g est symétrique en 0 (c'est-à-dire $g(-t) = g(t)$), qui est d'ailleurs l'algorithme original proposé par Metropolis, Rosenbluth, Rosenbluth, Teller & Teller (1953).

Algorithme 1.3.
Étant donné x_t;

1. *Générer $y_t \sim g(y - x_t)$;*
2. *prendre*

$$x_{t+1} = \begin{cases} y_t & \textit{avec probabilité} \quad \min\left\{\frac{f(y_t)}{f(x_t)}, 1\right\} \\ x_t & \textit{Sinon} \end{cases}$$

Exemple 1.7. Nous reprenons le même exemple, cette fois on simule $\Gamma(2,4)$ à partir d'une loi uniforme $Y_t \sim U(X_t - 4,\, X_t + 4)$.

Programme 1.6.

```
N=100000; a=2; b=4; bn=30
x=numeric(1); x[1]=0.5;
f=function(x){((a^b)/gamma(a))*(x^(a-1))*exp(-b*x)}
for(i in 2:N){
y=runif(1,x[i-1]-4,x[i-1]+4)
if(runif(1)<min(f(y)/f(x[i-1]),1)) {x[i]=y}
else {x[i]=x[i-1]} }
par(mfrow=c(1,2))
hist(x,breaks=bn,prob=TRUE,main="L'algorithme M-H par marche aléatoire")
curve(f,add=TRUE,col="blue",lty=1,lwd=2)
hist(rgamma(N,a,b),breaks=bn,prob=TRUE,main="Echantillon de la loi gamma")
curve(f,add=TRUE,col="blue",lty=1,lwd=2)
```

On remarque également que les deux distributions sont approches.

1.4 Méthodes MCMC (2): l'échantillonnage de Gibbs

La technique de Metropolis-Hastings présentée dans la section précédente est intéressante à cause de son universalité, mais, d'un autre côté, le manque de connexion entre le mécanisme de la loi de proposition q et la loi cible π peut être néfaste pour les propriétés de convergence de la méthode et, dans la pratique, peut facilement empêcher la convergence si la probabilité d'atteindre des parties éloignées du support de

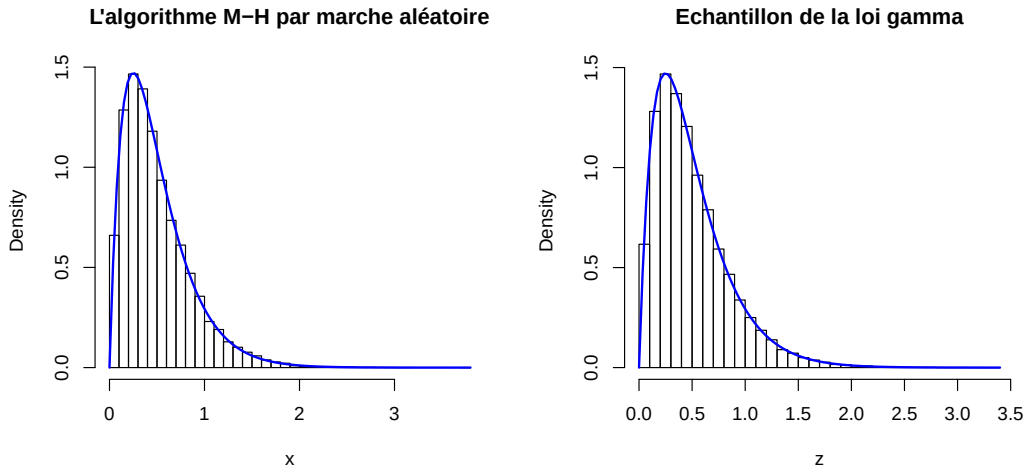

FIG 1.8. Histogramme pour les deux échantillons

la loi π est trop petite. L'approche de l'échantillonnage de Gibbs, repose sur une perspective différente. Cette méthode tire son nom des champs aléatoires de Gibbs, où elle a été utilisée pour la première fois par Geman et Geman (1984). La méthode de Gibbs est essentiellement basée sur les distributions conditionnelles comme les méthodes de modélisation bayésiennes.

1.4.1 Echantillonneur de Gibbs à deux étapes

L'échantillonneur de Gibbs à deux étapes crée une chaîne de Markov à partir d'une loi jointe donnée de la manière suivante. Si deux variables aléatoires X et Y ont pour densité jointe $f(x,y)$, avec les densités conditionnelles correspondantes $f_{Y|X}$ et $f_{X|Y}$, alors l'échantillonneur de Gibbs à deux étapes génère une chaîne de Markov (X_t,Y_t) selon les étapes suivantes :

Algorithme 1.4. *(Echantillonneur de Gibbs à deux étapes)*
Prendre $X_0 = x_0$ *;*
Pour $(t = 1$ *jusqu'à* $n)$ *générer*

1. $Y_t \sim f_{Y|X}(\cdot \mid x_{t-1})$;
2. $X_t \sim f_{X|Y}(\cdot \mid y_t)$.

Exemple 1.8. (loi normale bivariée)
On désire simuler une loi normale bivariée

$$(X,Y) \sim \mathcal{N}(0,\Lambda) \quad \text{avec} \quad \Lambda = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

Les lois conditionnelles:

$$Y|X = x \sim \mathcal{N}(\rho x, 1-\rho^2) \quad \text{et} \quad X|Y = y \sim \mathcal{N}(\rho y, 1-\rho^2)$$

L'implémentation en R

Programme 1.7.

```
N=20000; a=0.2; bn=40; mu=0;
f=function(x) {(1/sqrt(pi*2))*exp(-(1/2)*(x-mu)^2)}
y=x=numeric(1); y[1]=0.5;
x[1]=rnorm(1,a*y[1],1-a^2)
for(i in 2:N){
y[i]=rnorm(1,a*x[i-1],1-a^2)
x[i]=rnorm(1,a*y[i],1-a^2)}
par(mfrow=c(1,2))
hist(y,breaks=bn,prob=TRUE,main="Distribution de Y")
curve(f,add=TRUE,col="blue",lty=1,lwd=2)
hist(x,breaks=bn,prob=TRUE,main="Distribution de X")
```

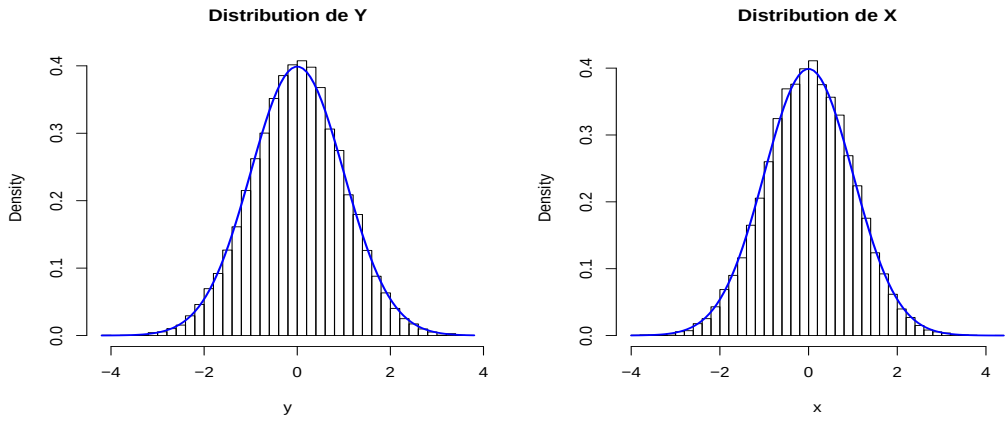

FIG 1.9. Histogramme pour deux distributions marginales d'une loi normale bivariée par échantillonneur de Gibbs

1.4.2 Echantillonneur de Gibbs à plusieurs étapes

Echantillonneur de Gibbs à plusieurs étapes est considéré comme une généralisation de l'échantillonneur de Gibbs à deux étapes. Supposons que, pour $p > 1$, la variable aléatoire $X \in \mathcal{X}$ puisse s'écrire comme $X = (X_1, \ldots, X_p)$, où les X_i sont des composantes à une ou plusieurs dimensions, si on peut calculer les densités conditionnelles correspondantes $f_1, f_2, \ldots, f_p$, alors l'implémentation devient facile.

Algorithme 1.5. *(Echantillonneur de Gibbs à plusieurs étapes)*
étant donné $x^{(t)} = (x_1^{(t)}, \ldots, x_p^{(t)})$*;*
Prendre $X^{(0)} = x^{(0)}$*;*
Pour $(t = 1$ *jusqu'à* $n)$ *générer*

1. $X_1^{(t+1)} \sim f_1(x_1 \mid x_2^{(t)}, \ldots, x_p^{(t)})$;
2. $X_2^{(t+1)} \sim f_2(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \ldots, x_p^{(t)})$;

$\vdots$

p. $X_p^{(t+1)} \sim f_p(x_p \mid x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{p-1}^{(t+1)})$.

Mesure invariante π:

Montrons que $\pi P = \pi$ dans le cas $k = 2$ (par souci de simplicité des écritures)
Le noyau de transition est donné par (sa densité)

$$P(x, y) = \prod_{i=1}^{k} \pi_i(\ \cdot\ \mid y_1, \ldots, y_{i-1}, x_{i+1}, \ldots, x_k).$$

Soit $B \in \mathcal{B}(\mathcal{X})$

$$\begin{aligned}
\int_{\mathcal{X}\times B} \pi(dx)P(x, dy) &= \int_{\mathcal{X}\times B} \pi(x_1, x_2)\pi_1(y_1|x_2)\pi_2(y_2|y_1)dxdy \\
&= \int_{\mathcal{X}\times B} \pi(x_1, x_2)\frac{\pi(y_1, x_2)}{\int \pi(z, x_2)dz}\frac{\pi(y_1, y_2)}{\int \pi(y_1, z)dz}dxdy \\
&= \int_{\mathcal{X}\times B} \frac{\pi(x_1, x_2)}{\int \pi(z, x_2)dz}\frac{\pi(y_1, x_2)}{\int \pi(y_1, z)dz}\pi(y_1, y_2)dxdy \\
&= \int_{\mathcal{X}\times B} \pi_1(x_1|x_2)\pi_2(x_2|y_1)\pi(y_1, y_2)dxdy \\
&= \int \pi(y)\left(\int_{\mathcal{X}} \pi_1(x_1|x_2)\pi_2(x_2|y_1)dx\right)dy \\
&= \pi(B).
\end{aligned}$$

Exemple 1.9. Modèle Auto-exponentiel de Besag (1974)
Soit une densité f, appelée auto-exponentielle définie comme suit

$$f(x_1,x_2,x_3) \propto \exp\big\{-(x_1 + x_2 + x_3 + \theta_1 x_1 x_2 + \theta_2 x_2 x_3 + \theta_3 x_1 x_3)\big\}$$

avec θ_1, θ_2, θ_3 connus.
Toutes les lois conditionnelles sont exponentielles:

$$X_1 \mid x_2,x_3 \sim Exp(1+\theta_1 x_2+\theta_3 x_3).$$
$$X_2 \mid x_1,x_3 \sim Exp(1+\theta_1 x_1+\theta_2 x_3).$$
$$X_3 \mid x_2,x_1 \sim Exp(1+\theta_2 x_2+\theta_3 x_1).$$

Nous prenons pour notre cas $\theta_1 = 0.1$, $\theta_2 = 2$ et $\theta_3 = 20$.

Programme 1.8.

```
N=20000; a=0.1; b=2; c=20; bn=40;
X1=X2=X3=numeric(1);
X3[1]=X2[1]=0.5;
for (i in 2:N) {
X1[i]=rexp(1,1+a*X2[i-1]+c*X3[i-1])
X2[i]=rexp(1,1+a*X1[i]+b*X3[i-1])
X3[i]=rexp(1,1+c*X1[i]+b*X2[i]) }
par(mfrow=c(2,2))
hist(X1,breaks=bn,prob=TRUE,main="Distribution de X1")
hist(X2,breaks=bn,prob=TRUE,main="Distribution de X2")
hist(X3,breaks=bn,prob=TRUE,main="Distribution de X3")
```

1.4.3 Données manquantes

Jusqu'ici on peut remarquer que l'échantillonneur de Gibbs ne génère que les vecteurs $X = (X_1, \ldots, X_p)$ avec $p > 1$ composantes, contrairement à l'algorithme de Metropolis-Hastings. Alors si on dispose d'une densité marginale $f_X(x)$, on peut construire (ou compléter $f_X(x)$ en) une densité jointe correspondante $f(x,y)$ pour but d'aider à la simulation, où la seconde variable Y est considérée comme *variable auxiliaire*. Il existe de nombreux contextes où $f_X(x)$ peut être naturellement complétée en $f(x,y)$ et associée à un échantillonneur de Gibbs.
Si on pose

$$f(x) = \int_y g(x,y)dy,$$

avec $X|Y$ et $Y|X$ explicitement calculées, alors on peut appliquer un échantillonneur de Gibbs (Algorithme 1.4).

Exemple 1.10. Supposons qu'on veut générer une loi exponentielle $X \sim Exp(1)$ avec l'échantillonneur de Gibbs, alors la solution est d'utiliser le modèle auto-exponentiel

$$f(x,y) \propto \exp\{-(x+y+\theta xy)\},$$

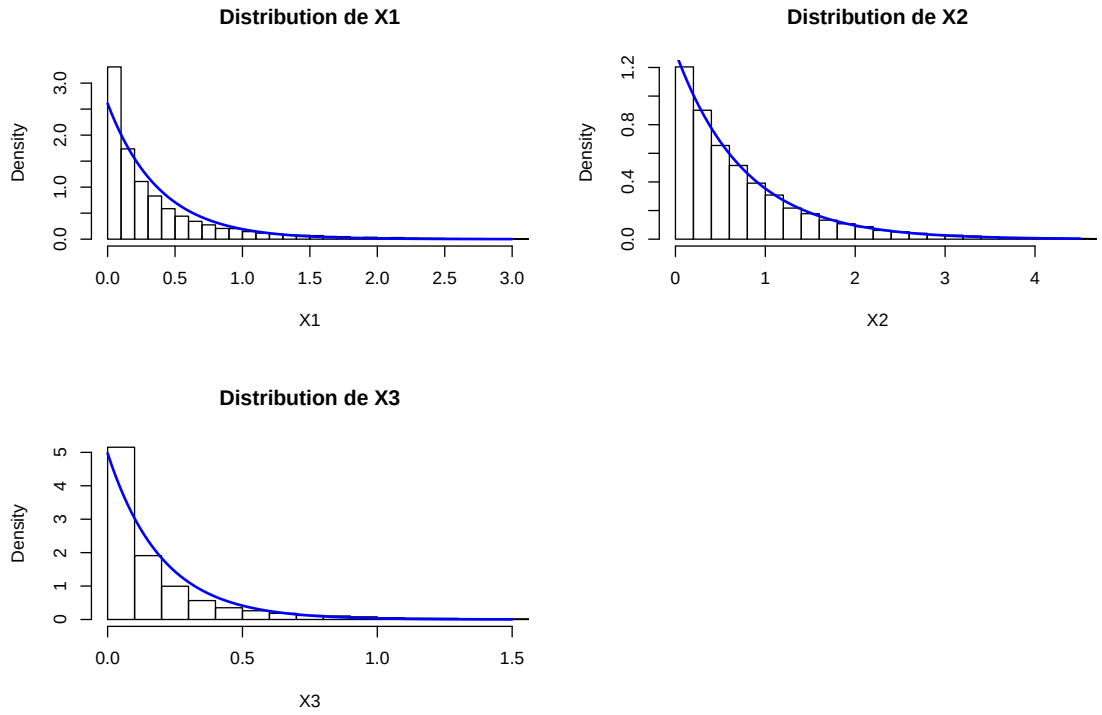

FIG 1.10. Histogramme de trois distributions du modèle auto-exponentiel

pour avoir facilement les lois conditionnelles suivantes

$$X|Y \sim Exp(1+\theta y)$$

$$Y|X \sim Exp(1+\theta x)$$

On prend $\theta = 3$, et nous donnons le programme R correspondant,

Programme 1.9.

```
N=20000; theta=3; bn=40;
Y=X=numeric(1); Y[1]=5;
for(i in 2:N){
X[i]=rexp(1,1+theta*Y[i-1])
Y[i]=rexp(1,1+theta*X[i])}
hist(X,breaks=bn,prob=TRUE,xlim=c(0,4),main="Distribution de X")
```

1.4.4 Méthodes hybrides

Les deux techniques Gibbs et M-H ne sont pas exclusives l'une de l'autre, mais on peut construire des algorithmes *hybrides*. C'est-à-dire, si dans un ensemble de lois conditionnelles complètes $f_1, \ldots, f_p$, une densité f_i (ou plus) ne peut pas être simulée

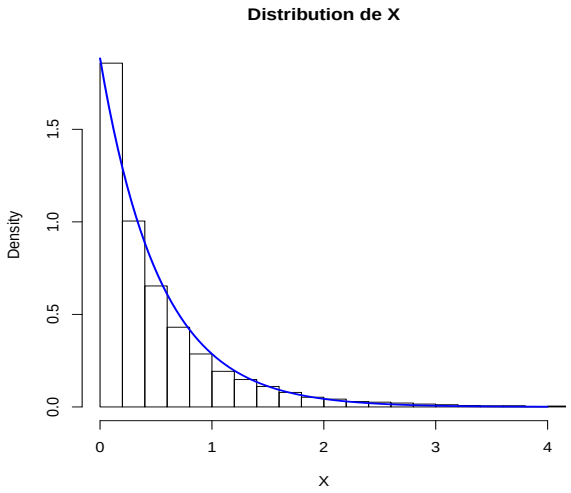

FIG 1.11. Histogramme une loi exponentielle générée par l'échantillonneur de Gibbs

directement, il est tout de même possible de construire un échantillonneur de Gibbs avec la stratégie d'intégrer une ou plusieurs étapes de l'algorithme M-H dans Gibbs.

Exemple 1.11. (loi normale bivariée)
On applique la méthode hybride pour la loi normale bivariée en supposant que la densité conditionnelle $f_{Y|X}$ est bien définie mais pas pour $f_{X|Y}$.

Programme 1.10.

```
N=50000; a=0.2; bn=40; mu=0;
g=function(x){(1/sqrt(pi*2))*exp(-(1/2)*(x-mu)^2)}
b=matrix(c(1,a,a,1),ncol=2); B=solve(b);
f=function(x,y){exp((-1/2)*(t(c(x,y))%*%B%*%(c(x,y))))}
y=x=numeric(1); y[1]=0.5;
x[1]=rnorm(1,a*y[1],1-a^2);
for (i in 2:N){
y[i]=rnorm(1,a*x[i-1],1-a^2)
u=runif(1,x[i-1]-4,x[i-1]+4)
if (runif(1)<min(f(u,y[i])/f(x[i-1],y[i]),1))
x[i]=u
else{x[i]=x[i-1]}}
par(mfrow=c(1,2));
hist(x,breaks=bn,prob=TRUE,main="Distribution de X")
curve(g,add=TRUE, col="blue",lty=1,lwd=2)
hist(y,breaks=bn,prob=TRUE,main="Distribution de Y")
```

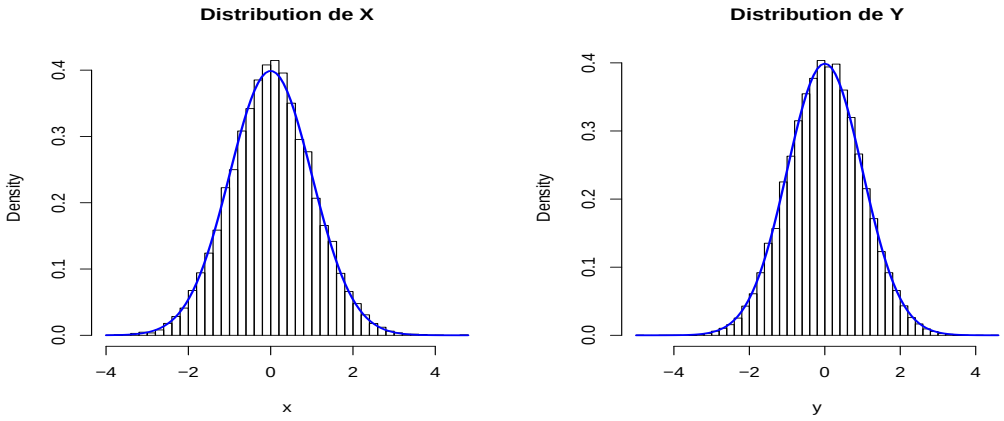

FIG 1.12. Histogramme pour deux distributions marginales d'une loi normale bivariée par la Méthode hybride

1.5 Algorithmes MCMC adaptatifs (AMCMC)

Un algorithme MCMC adaptatif est une extension des algorithmes MCMC développé récemment, où les noyaux de transition sont calibrés dynamiquement en fonction des performances observées jusqu'à l'itération courante.

1.5.1 Algorithme Metropolis Adaptatif (AM)

Haario et al. (2001)[3] ont développé une autre approche de l'algorithme Metropolis appelée adaptative, qui consiste à changer la distribution instrumentale $\mathcal{N}(x_i, \mathcal{C}_{i+1})$ pour chaque itération en mettant à jour la matrice de covariance $\mathcal{C}_{i+1}$ à chaque saut de la chaîne, et qui sera calculée à partir des valeurs antécédentes, x_1,$x_2, \ldots, x_i$.
En général, il est préférable de générer les premières i_0 valeurs à l'aide d'une distribution instrumentale fixe et les valeurs subséquentes à l'aide d'une distribution mise à jour selon la méthode décrite.
On commence par une loi cible d-dimensionnelle et on définit pour $i^{\text{ème}}$ itération la matrice de covariance $\mathcal{C}_i$ comme suit

$$\mathcal{C}_i = \begin{cases} \mathcal{C}_0 & \text{si } i \leq i_0; \\ \alpha Cov(X_0, \ldots, X_{i-1}) + \epsilon\, \alpha\, I_d & \text{si } i > i_0. \end{cases}$$

Le terme $\mathcal{C}_0$ représente la matrice de covariance fixée et i_0 indique le temps à partir duquel l'algorithme adaptatif est appliqué. Le paramètre α est un facteur multiplicatif qui dépend de la dimension et qui peut être optimisé (généralement donné par $\alpha = 2.38^2/d$) avec d la dimension. Un élément $\epsilon > 0$ est introduit afin d'assurer la non singularité de la matrice $\mathcal{C}_i$ et afin d'assurer la convergence théorique de l'algorithme (généralement $\epsilon = 0.001$). I est la matrice identité.

Pour la nouvelle valeur, elle sera donc acceptée avec probabilité :

$$\rho = \min\left\{1, \frac{\pi(y_{i+1})}{\pi(y_i)}\right\}.$$

Pour éviter les calculs répétés pour la matrice de covariance et la moyenne dans l'implémentation, alors il est préférable d'utiliser les relations de récurrence suivantes

$$\mathcal{C}_{n+1} = \frac{n-1}{n}\mathcal{C}_n + \frac{\alpha}{n}\Big(n\bar{X}_{n-1}\bar{X}'_{n-1} + (n+1)\bar{X}_n\bar{X}'_n + X_nX'_n + \epsilon\, I_d\Big)$$

et

$$\bar{X}_{n+1} = \frac{1}{n+1}\Big(n\bar{X}_n + X_{n+1}\Big)$$

Exemple 1.12. On repend toujours l'exemple de la loi normale bivariée

$$(X,Y) \sim \mathcal{N}(0, \Lambda) \quad \text{avec } \Lambda = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

3. Disponible sur: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.8948&rep=rep1&type=ps

On pose $\rho = 0.2$

Le programme R correspondant

Programme 1.11.

```
N=200000;  a=0.2;  bn=30;
d=2; A=(2.38^2)/d; AA=(0.1^2)/8
b=matrix(c(1,a,a,1),ncol=2)
B=solve(b);
f=function(y){exp((-1/2)*(t(y)%*%B%*%y))}
q=function(y,m){exp((-1/2)*(t(y-m)%*%C%*%(y-m)))}
I=matrix(c(1,0,0,1),ncol=2)
y=x=numeric(1); y[1]=x[1]=0.5; C=Co=b;
M3=M2=M=c(x[1],y[1]);
for (i in 2:N){
M3=M2; M2=M;
H=c(x[i-1],y[i-1])
M=(1/(i))*((i-1)*M+c(x[i],y[i]))
C=((i-2)/(i-1))*C+(A/(i-1))*((i-1)*M3%*%t(M3)-i*M2%*%t(M2)+H%*%t(H)+AA*I)
if(i>10000) {Co=C}
z=rnorm(2,H,A*Co+A*AA*I)
xx=c(x[i],y[i])
if (runif(1)<min(f(z)/f(H),1)){
x[i]=z[1]
y[i]=z[2]}
else { x[i]=x[i-1]
y[i]=y[i-1]}}
par(mfrow=c(1,2))
hist(y,breaks=bn,main="Distribution de Y")
hist(x,breaks=bn,main="Distribution de X")
```

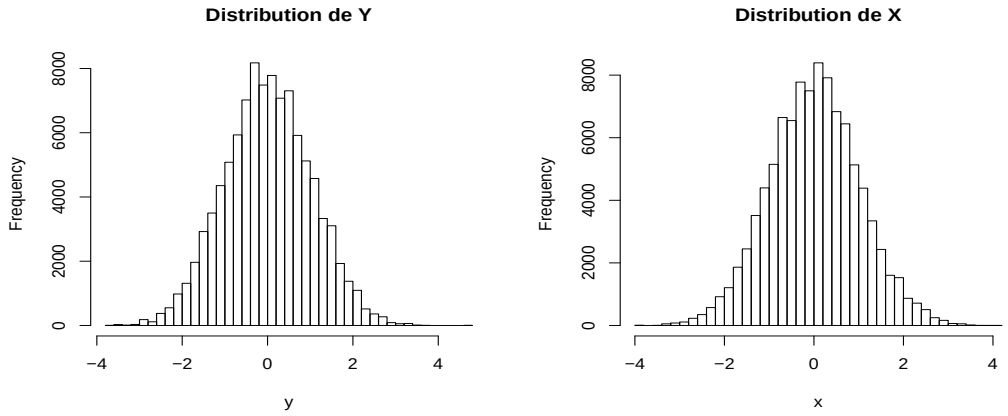

FIG 1.13. Histogramme pour deux distributions marginales d'une loi normale bivariée par AMCMC

On voir que l'adéquation avec les lois normales est remarquable.

Inconvénient

Malgré les améliorations apportées par les méthodes MCMC adaptatives, Robert et Casella (2004) soulignent que ces méthodes sont aussi critiquées. Bien que les résultats donnés par ces méthodes sont des fois satisfaisants mais parfois désastreux. En effet, si on calibre l'algorithme en fonction de sa sortie jusqu'à l'itération présente, cela signifie que l'algorithme n'est plus markovien puisqu'il dépend de tout le passé de la simulation.

1.6 La librairie coda

Un outil important qui va nous servir pour la programmation avec le langage R dans la suite est la librairie coda, développé par Plummer et al. (2006), qu'est jugé comme un outil performant et facile à l'implémenter. La librairie met en disposition plusieurs fonctions permettant de faire un diagnostic sur la convergence des chaînes produites par les méthodes MCMC. On peut charger toutes les fonctions coda grâce à la commande `library(coda)`. Elle contient un paquet de fonctions essentiel appelé mcmc. Pour pouvoir alors utiliser ces fonctions on doit transformer un vecteur `x` en un objet *mcmc* avec l'instruction `mcmc(x)`, ou bien `mcmc.list(mcmc(x[,1]),mcmc(x[,2]),...,mcmc(x[,T]))` si `x` est une matrice(T,n), qui va nous servir pour représenter des exécutions parallèles d'une même chaîne.

1.7 Contrôle de convergence des algorithmes MCMC

Parmi les points forts des méthodes MCMC est la garantie théorique de convergence de ces algorithmes car les chaînes qu'ils produisent sont ergodiques. Bien que ces résultats soient évidemment nécessaires en tant que validation théorique des algorithmes MCMC, ils sont toutefois insuffisants du point de vue de l'implémentation, pour une raison simple est : "*à quelle étape nous pouvons assurer que le nombre d'itérations est suffisant pour arrêter l'algorithme?*". On distingue ici deux notions de convergence, le premier type est la convergence vers la loi stationnaire, en effet, il s'agit de déterminer si la variable X_t est effectivement distribuée suivant la distribution stationnaire f. Le second type de convergence le plus important dans la mise en œuvre des algorithmes MCMC est la convergence de la moyenne empirique

$$\frac{1}{T}\sum_{t=1}^{T} h(X_t) \tag{1.21}$$

vers $\mathbb{E}_f\big[h(X)\big]$ pour une fonction arbitraire h. Le théorème ergodique fonde la convergence de cette moyenne d'un point de vue théorique, mais il s'agit à ce point de

déterminer la valeur minimale de T autorisant l'approximation $\mathbb{E}_f[h(X)]$ par (1.21). A propos de ce sujet de nombreuses méthodes ont été proposées dans la littérature, voir par exemple, Robert C.P. (1996, 1998, 2011), Gelman A. et Rubin D. (1992), Raftery, A.E. et Lewis, S. (1992a, 1992b) et Gilks W.R et al. (1996).

1.7.1 Densité (histogramme) des valeurs générées

Une idée tout à fait naturelle et celle utilisée jusqu'à présent. Elle s'agit en fait de comparer l'histogramme des valeurs générées par un algorithme MCMC avec la densité de la loi cible si elle est connue, sinon l'idée reste inutile!
Pour l'exemple (voir les sections précédentes).

1.7.2 Moyennes cumulées

Le principe de cette méthode est très simple, il consiste à suivre (visuellement) l'évolution des moyennes cumulées en fonction de t sur le graphe.

$$S_T = \frac{1}{T}\sum_{t=1}^{T} h(x_t)$$

Si la courbe des moyennes cumulées n'est pas stabilisée après T itérations, alors on doit augmenter le nombre d'itération.

Exemple 1.13. Supposons qu'on cherche à calculer $\mathbb{E}[X]$ tel que $X \sim Exp(\lambda)$ avec $\lambda = 1$, alors nous proposons d'utiliser l'algorithme de M-H par marche aléatoire en contrôlant la convergence avec cette méthode.

Programme 1.12.

```
T=10000; a=1;
fn=function(x){if (x<0) {0} else {a*exp(-a*x)}}
x=numeric(1); x[1]=0.6;
for (i in 2:T){
y=runif(1,x[i-1]-4,x[i-1]+4)
if (runif(1)<min(fn(y)/fn(x[i-1]),1)) {x[i]=y}
else{x[i]=x[i-1]}}
Somcum=cumsum(x)/(1:T)
plot(Somcum,type="l",ylab="La somme cumulée",xlab="Nombre d'itération")
abline(a=1/a,b=0,col="red")
```

On remarque que le graphe à gauche est instable pour un nombre d'itération $T = 10^4$, alors que le graphe à droite apparaît plus stable pour $T = 10^5$.

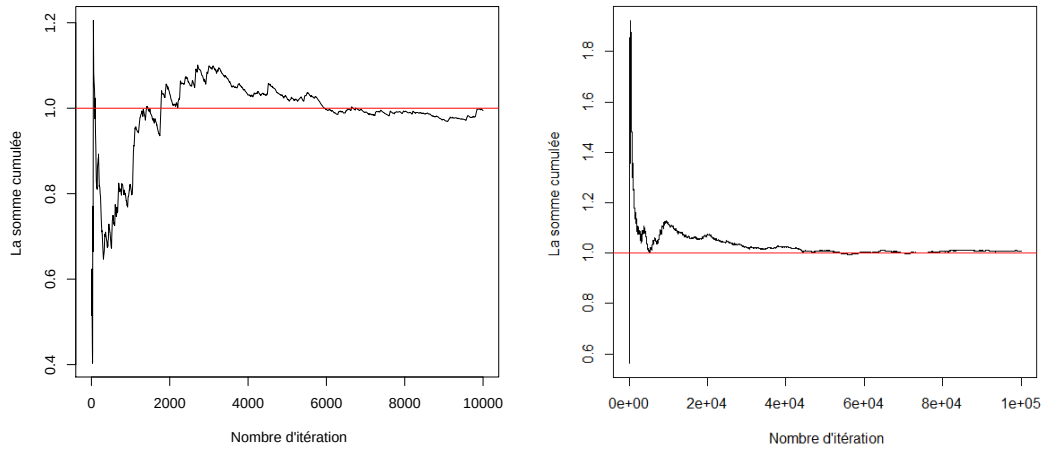

FIG 1.14. Evolution des moyennes cumulées avec $T = 10^4$ à gauche et $T = 10^5$ à droite

Inconvénient

Cette méthode n'est pas un bon outil permettant de répondre rigoureusement sur la question de convergence (voir le détail dans la section; problèmes de convergence [1.8.1]).

1.7.3 Contrôle binaire

Raftery et Lewis (1992a,b) [4] proposent une autre méthode qui sépare la chaîne $(X^{(t)})$ en deux états, ils définissent une chaîne

$$\xi^{(t)} = \mathbb{1}_{x^{(t)} \leq \tau}$$

où τ est choisi arbitrairement dans le support de π.
En utilisant une approximation markovienne de la loi de $(\xi^{(t)})$, avec:

$$P(\xi^{(t)} = 1 \mid \xi^{(t-1)} = 0) = \alpha$$

$$P(\xi^{(t)} = 0 \mid \xi^{(t-1)} = 1) = \beta.$$

Conduit à la distribution stationnaire

$$P(\xi^{(\infty)} = 0) = \frac{\beta}{\alpha + \beta}\ ; \qquad P(\xi^{(\infty)} = 1) = \frac{\alpha}{\alpha + \beta}$$

On peut donc déterminer la taille d'initialisation en imposant

$$\left| P(\xi^{(t_0)} = i \mid \xi^{(0)} = j) - P(\xi^{(\infty)} = i) \right| < \varepsilon \quad \text{pour} \quad i, j = 0, 1.$$

4. Disponible sur: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.6352&rep=rep1&type=ps

Les auteurs déduisent une évaluation du *temps de chauffe*, t_0, nécessaire pour approcher la loi stationnaire à ε près avec

$$t_0 \geq \frac{\log\left(\frac{(\alpha+\beta)\varepsilon}{\alpha\vee\beta}\right)}{\log|1-\alpha-\beta|}. \tag{1.22}$$

La taille de l'échantillon garantissant la convergence de

$$\delta_T = \frac{1}{T}\sum_{t=t_0}^{t_0+T} \xi^{(t)} \text{ vers } \frac{\alpha}{\alpha+\beta}$$

peut être déterminée par une approximation normale de δ_T, de variance

$$\frac{1}{T}\frac{(2-\alpha-\beta)\alpha\beta}{(\alpha+\beta)^3}.$$

Si on impose que

$$P\Big(\Big|\delta_T - \frac{\alpha}{\alpha+\beta}\Big| < q\Big) \geq \varepsilon',$$

D'où la valeur de T obtenue par Raftery et Lewis (1992, 1996) de la formule

$$\phi\left(\sqrt{T}\frac{(\alpha+\beta)^{3/2}q}{\sqrt{\alpha\beta(2-\alpha-\beta)}}\right) \geq \frac{\varepsilon'+1}{2},$$

est

$$T \geq \frac{(\alpha\beta(2-\alpha-\beta)}{q^2(\alpha+\beta)^3}\phi^{-1}\Big(\frac{\varepsilon'+1}{2}\Big). \tag{1.23}$$

Où ϕ est la fonction de répartition de la loi normale standard.

Exemple 1.14. (Suite de l'exemple précédant)
On applique cette méthode pour l'exemple précédant à fin de calculer le temps qu'il faut pour avoir la stationnarité.

Programme 1.13.

```
T=20000; a=1; const=2; eps=10^(-4);
fn=function(x){if (x<0) {0} else {a*exp(-a*x)}}
v=x=numeric(1); x[1]=1; b=c=0;
for(i in 2:T){
y=runif(1,x[i-1]-4,x[i-1]+4)
if (runif(1)<min(fn(y)/fn(x[i-1]),1)){x[i]=y}
else {x[i]=x[i-1]}
if (x[i]<const){v[i]=1}
else {v[i]=0}
if((v[i]==1) && (v[i-1]==0)) {b=b+1}
```

```
if((v[i]==0) && (v[i-1]==1)) {c=c+1}}
alpha=b/(T-1); beta=c/(T-1);
t=(log((alpha+beta)*eps)/max(alpha,beta))/log(abs(1-alpha-beta))
print(t)
Somcum=cumsum(x)/(1:T)
plot(Somcum,type="l",lwd=2,ylab="La somme cumulée",
     xlab="Nombre d'itération")
abline(a=1/a,b=0,col=2); abline(v=t,col=4)
```

La valeur donnée pour $\varepsilon = 10^{-4}$ est: $t = 5827$.

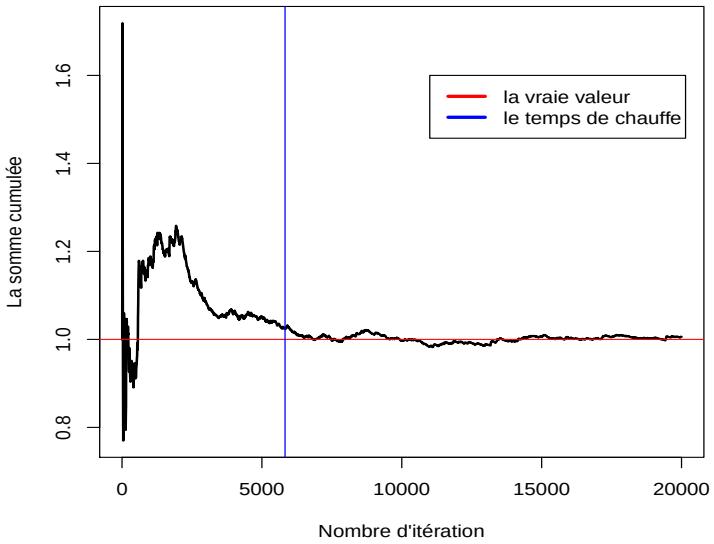

FIG 1.15. Temps de chauffe sur la courbe des moyennes cumulées

La figure [1.17] monte bien que la courbe commence à stabiliser effectivement après la 5827$^{\text{ème}}$ itération, d'où on peut dire que la chaîne simulée est devenue stationnaire à partir de cette valeur.

Inconvénient

Cette approche défendue par Raftery et Lewis soufre malheureusement de quelques inconvénients, en effet, la suite $(\xi^{(t)})$ proposée n'étant pas vraiment une chaîne de Markov! (voir Robert (1996) pour des critiques sur les fondements théoriques de ces approximations)

1.7.4 Variances intra et inter

La méthode de Gelman et Rubin (1992)[5] consiste à lancer plusieurs chaînes parallèles $(X_m^{(t)})_{t\in\mathbb{N}}$ avec $m = 1,\ldots,M$, éventuellement transformé en $\xi_m^{(t)} = h(X_m^{(t)})$, et les comparées.
On définit la variance inter-chaînes *(between)* comme la variance des moyennes

$$B_T = \frac{1}{M-1}\sum_{m=1}^{M}(\bar{\xi}_m - \bar{\xi})^2,$$

et la variance intra-chaîne *(within)* comme la moyenne des variances

$$W_T = \frac{1}{M-1}\sum_{m=1}^{M} s_m^2 = \frac{1}{M-1}\sum_{m=1}^{M}\frac{1}{T-1}\sum_{t=1}^{T}(\xi_m^{(t)} - \bar{\xi}_m)^2$$

avec

$$\xi_m^{(t)} = h(X_m^{(t)}), \qquad \bar{\xi}_m = \frac{1}{T}\sum_{t=1}^{T}\xi_m^{(t)}, \qquad \bar{\xi} = \frac{1}{M}\sum_{m=1}^{M}\bar{\xi}_m$$

Un premier estimateur de la variance *a posteriori* de ξ est

$$\sigma_T^2 = \frac{T-1}{T}W_T + B_T.$$

Gelman et Rubin (1992) comparent σ_T^2 et W_T , qui sont asymptotiquement équivalents à une approximation de Student. Les auteurs proposent donc un critère **PSRF** défini par

$$R_T = \frac{\sigma_T^2 + \frac{B_T}{M}}{W_T}\frac{\nu_T}{\nu_T - 2} \tag{1.24}$$

$$= \left(\frac{T-1}{T} + \frac{M+1}{M}\frac{B_T}{W_T}\right)\frac{\nu_T}{\nu_T - 2} \tag{1.25}$$

avec

$$\nu_T = \frac{(\sigma_T^2 + \frac{B_T}{M})^2}{W_T}$$

Alors si le critère **PSRF** est proche de 1, on peut conclure que chaque distribution est proche de la distribution de la loi cible.
En utilisant des approximations normales, la distribution de R_T peut être déduite de l'approximation

$$TW_T/B_T \sim \mathcal{F}(M-1,\vartheta_T), \quad \text{avec} \quad \vartheta_T = 2W_T^2/\omega_T,$$

et

$$\omega_T = \frac{1}{M^2}\left[\sum_{m=1}^{M} s_m^4 - \frac{1}{M}\Big(\sum_{m=1}^{M} s_m^2\Big)^2\right].$$

5. Disponible sur: http://www.stat.columbia.edu/ gelman/research/published/itsim.pdf

Par conséquent, la distribution de Ficher peut être utilisée pour tester l'égalité de R_T à 1 et construire des intervalles de confiance pour R_T.

Exemple 1.15. Suite de l'exemple (1.13)
On rappelle: $X \sim Exp(\lambda)$ avec $\lambda = 1$.
On lance 10 chaînes parallèles en utilisant l'algorithme M-H par marche aléatoire pour les générées.
Pour cette méthode la tâche est facile car la librairie `coda` nous fournit deux fonctions importantes, la première est: `gelman.diag` qui donne la dernière valeur du critère PSRF, la deuxième: `gelman.plot` pour représenter la figue de l'évolution de PSRF en fonction de nombre de simulation.

Le programme donc est comme suit

Programme 1.14.

```
N=2000; T=10; a=1; x=numeric(1);
Z=matrix(rep(0,N*T),nrow=N)
fn=function(x){if (x<0) {0} else {a*exp(-a*x)}}
for(j in 1:T){
x[1]=runif(1,0,5)
for(i in 2:N){y=runif(1,x[i-1]-2,x[i-1]+2)
if(runif(1)<min(fn(y)/fn(x[i-1]),1)) {x[i]=y}
else {x[i]=x[i-1]}}
Z[,j]=x}
library(coda)
library(lattice) # ce package est nécessaire pour le package coda
data=mcmc.list(mcmc(Z[,1]),mcmc(Z[,2]),mcmc(Z[,3]),mcmc(Z[,4]),mcmc(Z[,5]),
mcmc(Z[,6]),mcmc(Z[,7]),mcmc(Z[,8]),mcmc(Z[,9]),mcmc(Z[,10]))
gelman.plot(data,transform=TRUE,xlab="Nombre d'itération",ylab="PSRF")
gelman.diag(data)
```

Résultat

```
Potential scale reduction factors:

     Point est. Upper C.I.
[1,]       1.01       1.01
```

On remarque que le critère PSRF s'approche de 1 à partir de la 1500$^{\text{ème}}$ itération et donc la chaîne devient stationnaire à partir de cette valeur.

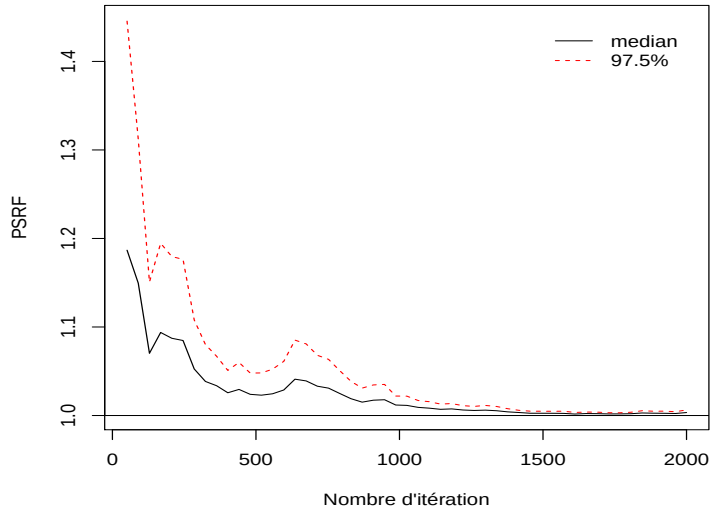

FIG 1.16. Evaluation de critère PSRF

1.7.5 Tests non paramétriques de stationnarité

Si nous souhaitons avoir plus de certitude sur la stationnarité de la chaîne de Markov $(X^{(t)})$ qu'une simple vérification graphique, nous devons vérifier, de manière statistique, que la distribution de la chaîne ne change pas au cours des itérations. Les tests non paramétriques standards d'adéquation, comme les tests de Kolmogorov-Smirnov et de Cramer-von, peuvent être appliqués à une réalisation unique de la chaîne $(X^{(t)})$. Comme les tests non paramétriques sont calibrés en termes d'échantillons iid, on doit cependant les corriger pour prendre en compte la corrélation entre les $X^{(t)}$. Pour ce faire on utilise la technique du *sous-échantillonnage (ou échantillonnage par lots)* pour réduire ou éliminer la corrélation entre les valeurs successives de la chaîne de Markov. Cette technique sous-échantillonne la chaîne $X^{(t)}$ avec un pas (déterministe ou aléatoire) k, en considérant seulement les valeurs $Y^{(t)} = X^{(kt)}$. Si la covariance $\mathbb{C}\text{ov}_f(X^{(0)},X^{(t)})$ décroît de manière monotone avec t, la justification du sous-échantillonnage est évidente. Mais dans certains cas la détermination de la valeur k reste une question délicate ou même impossible, c'est le cas où la covariance n'est pas décroissante ou si elle oscille avec t.

1.8 Problèmes de convergence

Dans l'implémentation des algorithmes MCMC, quelques questions sont très importantes comme: l'exploration du support de f (c'est-à-dire de la région qui contient la plupart de la masse de f), le degré de autocorrélation entre les X_t ainsi que le choix

de la loi de proposition, car ceci influencent directement sur la vitesse de convergence, alors il sera injuste de ne pas les prendre en compte, comme nous allons voir.

1.8.1 Exploration du support

En général les algorithmes MCMC souffrent d'un inconvénient majeur réside dans le fait que la partie du support de f qui n'a pas encore été visitée par la chaîne au temps T est presque impossible à détecter. Ce problème de la "*masse manquante*" est une difficulté centrale pour la plupart des algorithmes MCMC, beaucoup plus que le manque d'exploration des queues de distribution de f.

Exemple 1.16. On désire simuler une loi bêta : $X \sim Beta(\alpha, 1)$ avec $\alpha < 1$ à partir d'une loi $Beta(\alpha + 1, 1)$, puis on contrôle la convergence de l'estimateur de $\mathbb{E}(X)$.
La probabilité d'acceptance est

$$\rho = \min\left\{1, \frac{x^{\alpha-1}\theta^{\alpha}}{x^{\alpha}\theta^{\alpha-1}}\right\} = \min\{1, \theta/x\}$$

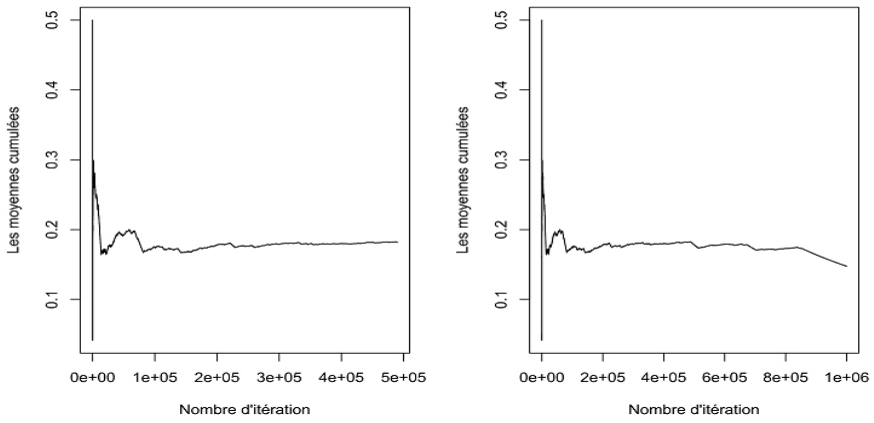

FIG 1.17. Les moyennes cumulées de même chaîne pour 5×10^5 itérations à gauche et un million à droite

Dans cet exemple si on contente d'un demi-million d'itérations alors nous allons constater d'après la figure [1.17] à gauche que la convergence est bien établie, mais ce n'est pas le cas comme le montre la figure à droite, ce problème revient, en fait, que la chaîne n'a pas encore explorer le support de la loi cible.

1.8.2 Le choix de la loi de proposition

Le choix de la loi de proposition q peut avoir un grand impact sur les performances de l'algorithme de Metropolis-Hastings. Si par exemple la chaîne déplace lentement ou

rapidement sur le support de la cible alors la chaîne aura des difficultés à explorer son support et donc elle convergera très lentement.

Exemple 1.17. supposons qu'on veut simuler une loi $\mathcal{N}(0,1)$ à partir d'une loi de proposition uniforme $U(x_t-\delta, x_t+\delta)$, avec δ différent.
Pour voir l'impact sur la loi cible, nous représentons trois figures en changeant le paramètre δ avec un nombre d'itération fixe.

Pour $\delta=3$: la figure [1.18] à droite nous indique que la chaîne déplace normalement (*ni lentement ni rapide*) sur le support de f, et à gauche on voit que l'échantillon produit par l'algorithme M-H coïncide avec la loi normale standard tracée en bleu.

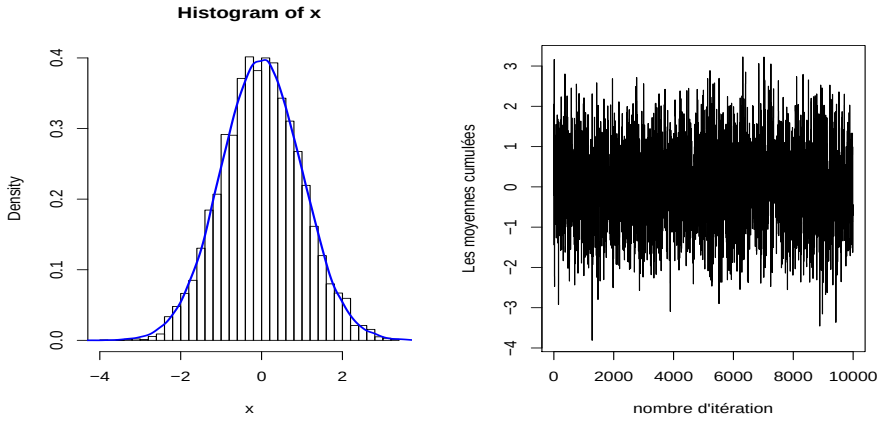

FIG 1.18. Une loi cible $\mathcal{N}(0,1)$ avec une loi de proposition $U(x_t-3, x_t+3)$

Pour $\delta=1$: on voit sur la figure [1.19] à droite que la chaîne déplace lentement sur le support de f, et à gauche une inadéquation de l'échantillon avec la loi normale standard.

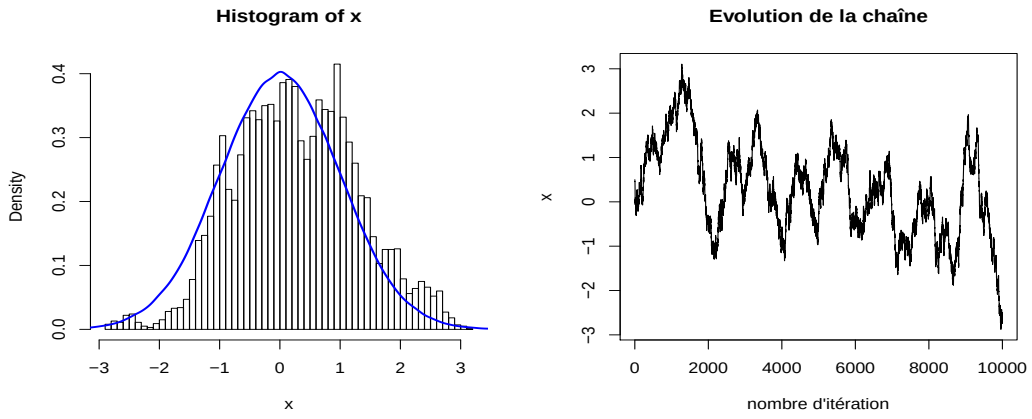

FIG 1.19. Une loi cible $\mathcal{N}(0,1)$ avec une loi de proposition $U(x_t-0.1, x_t+0.1)$

Pour $\delta = 50$: on remarque sur la figure [1.20] à droite que la chaîne déplace rapidement sur le support de f, ce qui provoque un long séjour presque dans chaque itération, et à gauche une inadéquation de l'échantillon avec la loi normale standard.

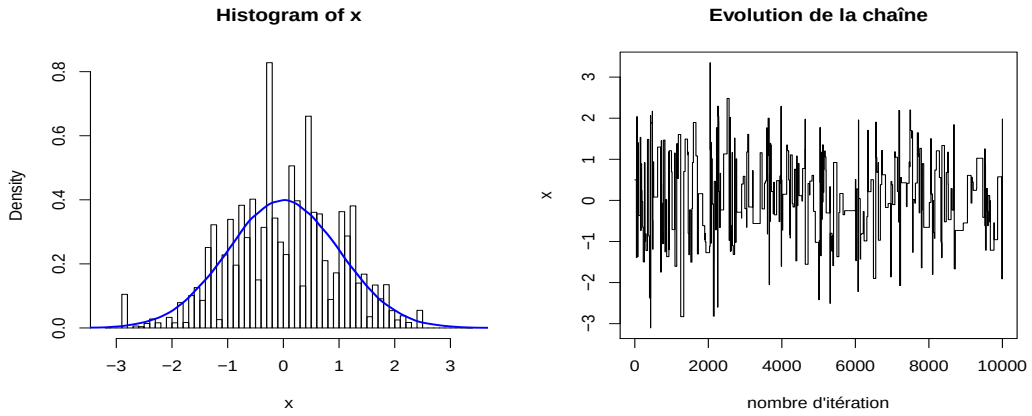

FIG 1.20. Une loi cible $\mathcal{N}(0,1)$ avec une loi de proposition $U(x_t-50, x_t+50)$

Sur la figure [1.21] on peut voir l'influence sur la convergence pour l'estimateur de l'espérance $\mathbb{E}(X)$.

Sur cette figure on voir bien que le déplacement long ou rapide de la chaîne agit

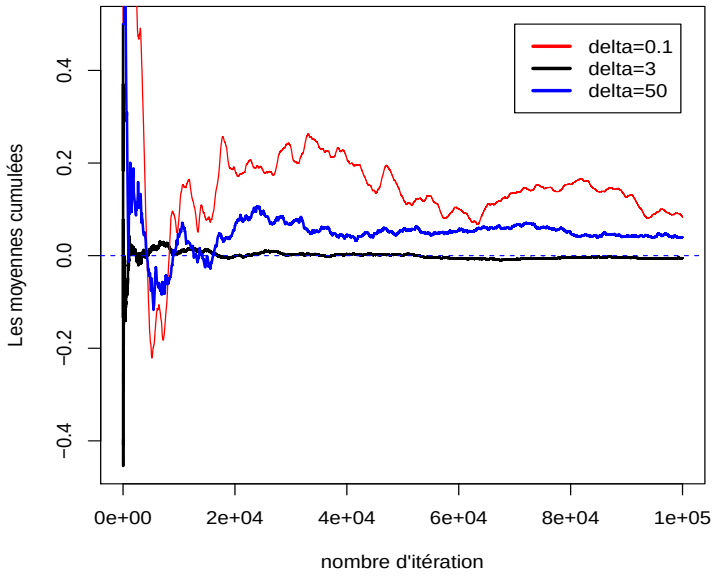

FIG 1.21. Trois estimations de $\mathbb{E}(X)$ avec les trois lois de proposition différentes

une façon significative sur la convergence des estimateurs.

1.8.3 La corrélation entre les valeurs successives de la chaîne

Un autre facteur qui peut provoquer un ralentissement pour la convergence vers la loi stationnaire est la corrélation entre les valeurs successives de la chaîne de Markov.

Exemple 1.18. (Suite de l'exemple [1.8]: Loi normale bivariée)
En utilisant l'échantillonneur de Gibbs pour générer une loi normale bivariée avec deux valeur différent de ρ.
On rappelle que la matrice de covariance $\Lambda = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$

On remarque sur la figure [1.22] que dans l'absence de corrélation, l'échantillon généré coïncide avec loi normale standard tracée en bleu.
Par contre, la figure [1.23] montre que si la corrélation est forte entre les valeurs successives de la chaîne alors l'échantillon généré sera éloigné de la loi cible.

1.8.4 La valeur initiale

Il existe parfois des situations où le choix de point de départ peut être aussi décisif, en effet, les algorithmes MCMC peuvent fournir des résultats tout à fait faux sur tout

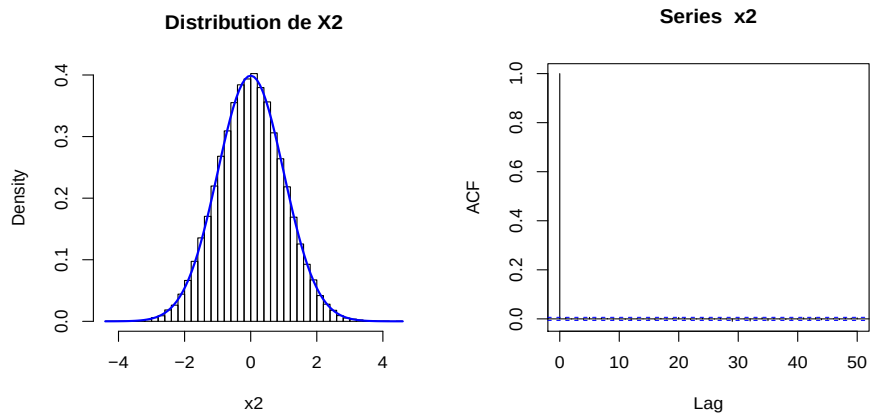

FIG 1.22. Histogramme et graphe de d'autocorrélation pour $\rho = 0.1$

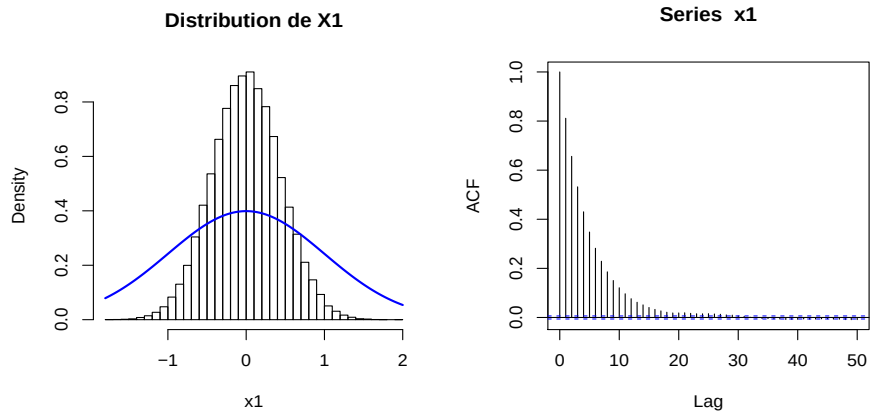

FIG 1.23. Histogramme et graphe de d'autocorrélation pour $\rho = 0.9$

si la chaîne déplace lentement sur le support de la loi cible, par conséquent, la chaîne reste beaucoup de temps dans la même région.

Exemple 1.19. (Un modèle de mélange gaussien)
Soit un échantillon de taille n d'une loi de mélange gaussien à deux composantes de même variance,

$$p\mathcal{N}(\mu_1, \sigma^2) + (1-p)\mathcal{N}(\mu_2, \sigma^2)$$

avec $p = 0.3, \quad \mu_1 = 0, \quad \mu_2 = 8$ et $\sigma^2 = 1$
On lance deux chaînes avec deux points de départ différents $x_0 = 0.5$ et $x_0 = 6$, puis nous représentons les résultats sur deux figures différentes.

Programme 1.15.

```
N=50000; a=2; b=4; bn=30; d=8;
x=numeric(1); x[1]=.5;
```

```
f=function(x,mu=0) {(1/sqrt(pi*2))*exp(-(1/2)*(x-mu)^2)}
g=function(x,mu=d) {(1/sqrt(pi*2))*exp(-(1/2)*(x-mu)^2)}
h=function(x,p=.3) {p*f(x)+(1-p)*g(x)}
for(i in 2:N){
y=runif(1,x[i-1]-.2,x[i-1]+.2)
if(runif(1)<min(h(y)/h(x[i-1]),1)) {x[i]=y}
else {x[i]=x[i-1]}}
hist(x,breaks=bn,xlim=c(-4,d+3),prob=TRUE,
main="Echantillon généré par l'algorithme M-H")
curve(h,add=TRUE,col="blue",lty=1,lwd=2)
```

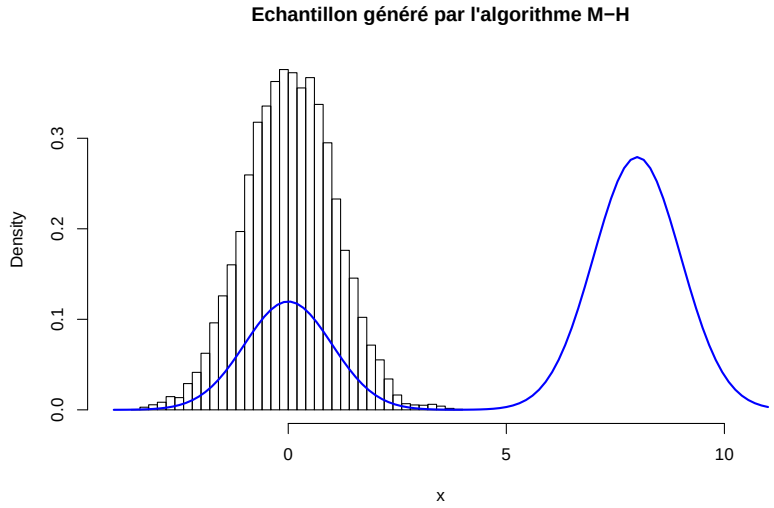

FIG 1.24. Histogramme d'un échantillon de mélange gaussien généré par M-H pour $x_0 = 0.5$

On remarque que si on se place pour la première valeur dans un mode alors on reste dedans pour des milliers de valeurs, et pour explorer tout l'espace ça nécessite donc un nombre d'itérations plus grand.

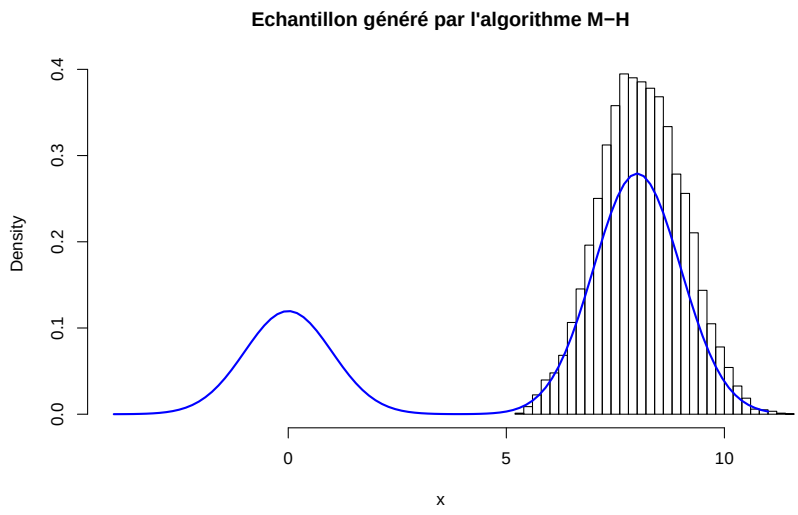

FIG 1.25. Histogramme d'un échantillon de mélange gaussien généré par M-H pour $x_0 = 6$

Chapitre 2

Méthodes de réduction de variance

2.1 Introduction

La réduction de variance reste un objet central pour beaucoup de chercheurs car il existe nombreuses situations où le calcul d'une quantité n'est pas facile à cause de sa variation, qui provoque une convergence très lente, par conséquent, les algorithmes qui visent à estimer cette valeur, notamment les méthodes de Monte Carlo ou MCMC, nécessitent parfois des millions d'itérations à fin d'avoir une approximation satisfaisante. Alors il existe des stratégies globales d'accélération, qui visent à exploiter les résultats de la simulation pour fournir des estimateurs alternatifs plus efficaces pour la même quantité.
Dans ce chapitre on présente quelques méthodes déjà proposées dans la littérature, voir par exemple Dagpunar (2007), Korn and Korn (2010), Philippe et Robert (2000), Rubinstein (2008) et Glasserman (2004).

2.2 Echantillonnage Préférentiel (E.P)

La méthode d'échantillonnage préférentiel est fondée sur une représentation alternative de

$$\mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x)f(x)dx. \tag{2.1}$$

Pour une densité arbitraire donnée g (dite *instrumentale*) strictement positive, avec $h \times f$ différent de zéro on peut réécrire (2.1) comme

$$\mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x)\frac{f(x)}{g(x)}g(x)dx = \mathbb{E}_g\left[h(X)\frac{f(X)}{g(X)}\right]; \tag{2.2}$$

qui devient une espérance sous la densité g et ce qui nous permet donc d'utiliser l'estimateur

$$\delta_{ep} = \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}h(X_i) \longrightarrow \mathbb{E}_f[h(X)], \tag{2.3}$$

fondé sur un échantillon $X_1, \ldots, X_n$ généré par g, avec une contrainte importante sur le support de g, est que $supp(g) \supset supp(h \times f)$.

Le choix de la fonction instrumentale g pour que la variance de δ_{ep} soit minimale est donné par le théorème suivant :

Théorème 2.1. *(Robert and Casella 2004)*
Le choix de g minimisant la variance de l'estimateur (2.2) est

$$g^*(x) = \frac{|h(x)|f(x)}{\int_{\mathcal{X}} |h(z)|f(z)dz}.$$

Démonstration 2.1. (*Voir la référence précédente page* 95).

Mais cette optimalité est seulement formelle, car cela oblige la connaissance de la quantité $\int_{\mathcal{X}} |h(z)|f(z)dz$, alors que c'est l'objet de l'étude elle même.

Exemple 2.1. On souhaite estimer l'espérance d'une loi exponentielle de densité $f(x) = 6e^{-6x}$,

$$\mathbb{E}_f[X] = \int 6xe^{-6x}dx \quad \text{par} \quad \delta = \sum_{i=1}^{n} X_i.$$

Nous proposons d'utiliser l'échantillonnage préférentiel pour construire un autre estimateur d'une variance réduite. Si on pose alors $g(x) = 3e^{-3x}$, donc on réécrit

$$\mathbb{E}_f[X] = \int \frac{6xe^{-6x}}{3e^{-3x}} 3e^{-3x}dx = \mathbb{E}_g\left[2Xe^{-3X}\right].$$

Qu'on l'estime également par

$$\delta_{ep} = \frac{1}{n}\sum_{i=1}^{n} 2X_i e^{-3X_i}.$$

Pour mieux voire l'effet de ce choix sur la variance, nous donnons le programme implémenté sur `R` ainsi que la figure résultante.

Le programme en `R` correspondant:

Programme 2.1.

```
n=3000; b=3;
f=function(x,a=6){if (x>0){a*exp(-a*x)} else {0}}
g=function(x,a=3){if (x>0){a*exp(-a*x)} else {0}}
fg=function(x){if (x>0) {x*f(x)/g(x)} else{0}}
x=rexp(n,6)
y=fg(rexp(n,b))
estx=cumsum(x)/(1:n)
```

```
bornes=sqrt(cumsum((x-estx)^2))/(1:n)
plot(estx,type="l",xlab="Moyenne et variation",lwd=2,ylim=c(.12,.25))
lines(estx+2*bornes,col=1,lwd=1,lty=2)
lines(estx-2*bornes,col=1,lwd=1,lty=2)
esty=cumsum(y)/(1:n)
bornes=sqrt(cumsum((y-esty)^2))/(1:n)
lines(esty, type="l",col="red",xlab="Moyenne et variation",lwd=2)
lines(esty+2*bornes,col=2,lwd=1,lty=2)
lines(esty-2*bornes,col=2,lwd=1,lty=2)
legend(1500,.24,c("Estimateur ordinaire","Estimateur E.P"),
lwd=3, col=c("black","red"))
```

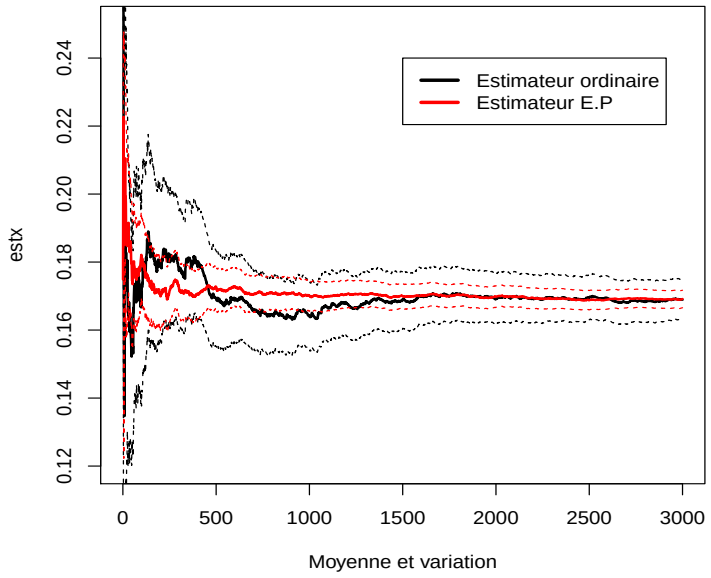

FIG 2.1. Evolution de deux estimateurs δ et δ_{ep} en fonction de nombre d'itération

On remarque sur figure [2.1] que la variance de l'estimateur δ_{ep} tracé en rouge avec l'intervalle de confiance à 95% est bien réduite en comparaison avec la variance de l'estimateur δ tracé en noir.

Nous répétons $T = 100$ expériences indépendantes pour un nombre de simulation n fixé, et nous calculons le facteur $Var(\delta)/Var(\delta_{ep})$, en estimant ces variances par

$$\frac{1}{T-1}\sum_{i=1}^{T}\left[\delta^{(i)}-\bar{\delta}\right]^2 \quad \text{et} \quad \frac{1}{T-1}\sum_{i=1}^{T}\left[\delta_{ep}^{(i)}-\bar{\delta}_{ep}\right]^2,$$

avec $\bar{\delta}$ la moyenne des $\delta^{(i)}$ et $\bar{\delta}_{ep}$ la moyenne des $\delta_{ep}^{(i)}$ respectivement. Les résultats que

nous avons obtenus sont présentés dans le tableau suivant:

Facteurs de réduction de variance					
	Nombre de simulation				
	$n = 1000$	$n = 10000$	$n = 50000$	$n = 100000$	$n = 200000$
Facteurs	5.31	5.44	5.56	5.50	5.16

Tab 2.1. Facteurs de comparaison entre δ_{ep} et δ

On remarque dans ce tableau que la variance de l'estimateur δ et environ 5 fois plus grande par rapport à la variance de l'estimateur δ_{ep} pour les différents nombres de simulation. On peut dire donc que le résultat fourni par l'estimateur δ_{ep} est 5 fois plus précis par rapport à l'estimateur δ, ou même, du point de vue du temps, si, par exemple, l'estimateur δ fournit un résultat en 5 heures, l'estimateur δ_{ep} le donne également en une heure avec la même précision.

Inconvénient

La liberté de choisir la loi instrumentale facile à simuler n'est pas un avantage quand le problème s'agit de réduire la variance, car ce choix ne sera pas généralement commode et la question va être plus compliquée, en effet, un choix arbitraire pour cette loi, conduit presque souvent à des fins non souhaitables, car il peut amplifier la variance à la place de la réduire, comme nous allons voir dans cet contre exemple.

Exemple 2.2. Si on reprend l'exemple précédant en changeant la loi instrumentale $Exp(3)$ par $Exp(10)$, alors le résultat est totalement changé.
La figure [2.2] montre bien que la variance de l'estimateur E.P en rouge est devenue plus grande.

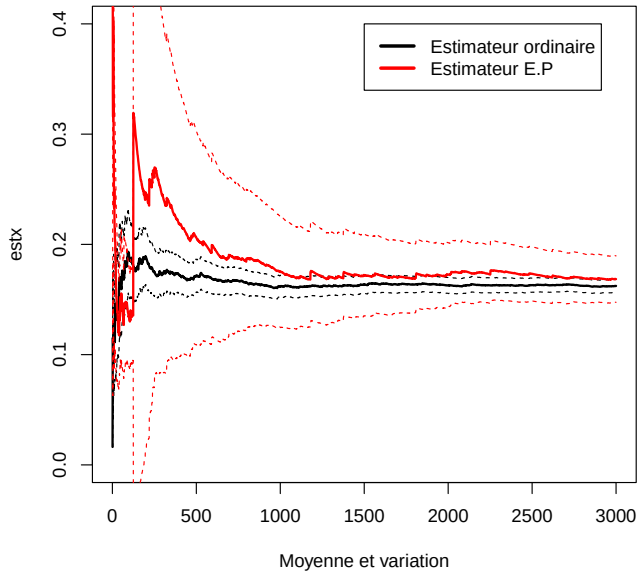

FIG 2.2. Evolution de deux estimateurs δ et δ_{ep} en fonction de nombre d'itération

Le tableau correspondant

Facteurs de réduction de variance				
	Nombre de simulation			
	$n = 1000$	$n = 10000$	$n = 50000$	$n = 100000$
Facteurs	0.055	0.046	0.044	0.046
$\frac{1}{\text{Facteurs}}$	18.16	21.3	22.7	21.72

Tab 2.2. acteurs de comparaison entre δ_{ep} et δ

D'après le facteur $\frac{1}{\text{Facteurs}}$, la variance de l'estimateur δ_{ep} est devenue de 18 à 22 fois plus grande par rapport à la variance de l'estimateur δ.

Remarque.
En pratique, pour éviter ce problème, on doit choisir une loi instrumentale g qui possède des queues plus lourdes que celles de f.

2.3 Variables antithétiques

Bien que les méthodes usuelles de simulation produisent des échantillons iid (ou quasi-iid) il est préférable parfois pour estimer une quantité de créer des échantillons de variables corrélées, car ceux-ci peuvent diminuer la variance de l'estimateur résultant.

La méthode des variables antithétiques est une procédure de réduction de variance la plus facile. Elle est basée sur l'idée de combiner un choix aléatoire des points avec systématiques, et qui exploite la présence de la corrélation pour réduire la variance. Supposons qu'on cherche à calculer

$$\mathcal{I} = \int_a^b h(x)dx. \tag{2.4}$$

D'après ce qu'on a vu dans la section (1.2), si U est une v.a uniforme sur $[a, b]$ alors on peut écrire (2.4) comme une espérance $\mathcal{I} = \mathbb{E}[h(U)]$. Dans ce cas la méthode de Monte Carlo usuelle ferait approximer $\mathcal{I}$ par

$$\delta_{mc} = \frac{1}{2n}\sum_{i=1}^{2n} h(U_i). \tag{2.5}$$

Avec la méthode de variables antithétiques on emploi également les variables U_1,$1-U_1,\ldots,U_n, 1-U_n$ pour introduire l'estimateur antithétique de Monte Carlo

$$\delta_{an} = \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n} h(U_i) + \frac{1}{n}\sum_{i=1}^{n} h(1-U_i)\right) \tag{2.6}$$

On note que U et $1-U$ ont la même distribution, les deux sommes sont des estimateurs sans biais pour $\mathbb{E}[h(U)]$. Par conséquent, l'estimateur antithétique est également sans biais.

Soit $\sigma^2 = \mathbb{V}\mathrm{ar}[h(U)]$, donc d'une façon évidente

$$\mathbb{V}\mathrm{ar}(\delta_{mc}) = \mathbb{V}\mathrm{ar}\left(\frac{1}{2n}\sum_{i=1}^{2n} h(U_i)\right) = \frac{1}{2n}\sigma^2$$

D'autre part, la variance de l'estimateur δ_{mc} est donnée par

$$\begin{aligned}
\mathbb{V}\mathrm{ar}(\delta_{an}) &= \frac{1}{n}\mathbb{V}\mathrm{ar}\Big[\frac{1}{2}\big(h(U)+h(1-U)\big)\Big] \\
&= \frac{1}{4n}\Big[\mathbb{V}\mathrm{ar}\big(h(U)\big) + \mathbb{V}\mathrm{ar}\big(h(1-U)\big) + 2\mathbb{C}\mathrm{ov}\big(h(U), h(1-U)\big)\Big] \\
&= \frac{\sigma^2}{2n} + \frac{1}{2n}\mathbb{C}\mathrm{ov}\Big(h(U),\, h(1-U)\Big)
\end{aligned}$$

Par conséquent, si $h(U)$ et $h(1-U)$ sont négativement corrélées,
c'est-à-dire, si $\mathbb{C}\mathrm{ov}\big(h(U),\, h(1-U)\big) < 0$ alors

$$\mathbb{V}\mathrm{ar}(\delta_{an}) < \mathbb{V}\mathrm{ar}(\delta_{mc}).$$

Généralisation

L'introduction des variables antithétiques est non seulement limitée aux variables aléatoires uniformes, mais elle peut se généraliser s'il existe une transformation $\mathcal{T}$: $\mathbb{R} \longrightarrow \mathbb{R}$ tels que:

1. $\mathcal{T}(X)$ possède la même distribution avec X.
2. $\mathbb{C}\mathrm{ov}\Big(h\big(\mathcal{T}(X)\big), h(X)\Big) \leq 0.$

Les exemples pour ceci sont des distributions symétriques comme la distribution normale, alors pour $X_i \sim \mathcal{N}(\mu, \sigma^2)$, la variable antithétique appropriée est donnée dans ce cas par $\mathcal{T}(X_i) = 2\mu - X_i$.

Proposition 2.1. *(Korn R., Korn E. and Kroisandt G. 2010)* ***Inégalité de la covariance de Chebyschev***
Soit X une v.a réelle. Soient h, g deux fonctions non décroissantes avec $\mathbb{C}\mathrm{ov}\big(h(x), g(x)\big)$ finie. Alors nous avons:

$$\mathbb{E}[h(X)g(X)] \geq \mathbb{E}[h(X)]\ \mathbb{E}[g(X)] \tag{2.7}$$

En effet, en choisissant $g(x) = -h(1-x)$, cette proposition s'applique directement.

Démonstration 2.2. On suppose que les fonctions f et g sont croissantes par rapport à chaque argument ; le raisonnement est similaire dans le cas où elles sont décroissantes. Soit X et Y des variables aléatoires réelles ; la croissance de f et g entraîne que

$$\mathbb{E}\Big[\big(f(X) - f(Y)\big)\big(g(X) - g(Y)\big)\Big] \geq 0.$$

On en déduit que

$$\mathbb{E}\big[f(X)g(X)\big] + \mathbb{E}\big[f(Y)g(Y)\big] \geq \mathbb{E}\big[f(X)g(Y)\big] + \mathbb{E}\big[f(Y)g(X)\big];$$

si on choisit une variable aléatoire Y indépendante de X et de même loi, alors on déduit que

$$\mathbb{E}[h(X)g(X)] \geq \mathbb{E}[h(X)]\ \mathbb{E}[g(X)]$$

Proposition 2.2. *(Korn R., Korn E. and Kroisandt G. 2010)*
Soit h une fonction non décroissante ou non croissante et soit X une v.a uniforme sur $[0, 1]$ avec $\mathbb{C}\mathrm{ov}\Big(h(X), h(1-X)\Big)$ finie. Alors nous avons:

$$\mathbb{C}\mathrm{ov}\Big(h(X), h(1-X)\Big) \leq 0. \tag{2.8}$$

En particulier, l'estimateur antithétique de Monte Carlo basé sur N nombres aléatoires possède une variance plus petite par rapport à l'estimateur de Monte Carlo basé sur $2N$ nombres aléatoires.

Exemple 2.3. Pour estimer l'intégrale suivante:

$$\mathcal{I} = \int_0^1 e^{-x} dx,$$

on utilise la méthode de Monte Carlo avec celle des variables antithétiques.
Soit alors $U \sim U[0,1]$.
Les estimateurs: Monte Carlo et antithétique de Monte Carlo s'écrivent donc

$$\delta_{mc} = \frac{1}{2n} \sum_{i=1}^{2n} exp(-u_i), \quad \text{et} \quad \delta_{an} = \frac{1}{2n} \sum_{i=1}^{n} \Big(exp(-u_i) + exp(1+u_i) \Big).$$

le code R est donné par

Programme 2.2.

```
n=20000; x=y=numeric(1);
f=function(x,a=1) {a*exp(-a*x)}
for(i in 1:n){
u=runif(1)
x[i]=(f(u)+f(1-u))/2}
y=f(runif(n))
m=cumsum(x)/(1:n)
bornes=sqrt(cumsum((x-m)^2))/(1:n)
plot(m,type="l",col=2,xlab="Nombre d'itération",
ylim=c(.62,.65),lwd=2,ylab="Les moyennes cumulées")
lines(m+2*bornes,col=2,lwd=1,lty=2)
lines(m-2*bornes,col=2,lwd=1,lty=2)
mm=cumsum(y)/(1:n)
bor=sqrt(cumsum((y-mm)^2))/(1:n)
lines(mm,type="l",col=1,lwd=2)
lines(mm+2*bor,col=1,lwd=1,lty=2)
lines(mm-2*bor,col=1,lwd=1,lty=2)
legend(10000,.65,c("Monte Carlo","variables antithétiques"),
lwd=3, col=c("black","red"))
```

Il est clair que la variance de l'estimateur δ_{an} en rouge est plus petite par rapport à la variance de l'estimateur δ_{mc} tracé en noir.

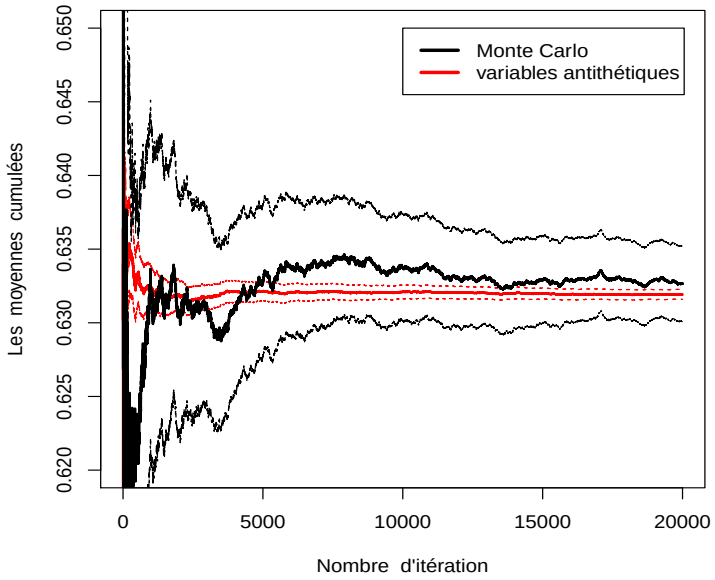

FIG 2.3. Evolution de deux estimateurs δ_{mc} et δ_{an} en fonction de nombre d'itération

Avec des simulations répétées de même expérience, nous obtenons le tableau des facteurs suivant:

Facteurs de réduction de variance					
	Nombre de simulation				
	$n = 1000$	$n = 10000$	$n = 50000$	$n = 100000$	$n = 200000$
Facteurs	41.49	44.86	54.02	61.78	73.14

Tab 2.3. Facteurs de comparaison entre δ_{an} et δ_{mc}

D'après les résultats du tableau le facteur de réduction de variance est assez grand surtout avec les premiers nombres de simulation (pour 1000 et 10000) par rapport aux autres méthodes, et on remarque aussi qu'il augmente avec le temps.

2.4 Variables de contrôle

La méthode des variables de contrôle est parmi les techniques les plus efficaces et les plus appliquées pour améliorer les performances des estimateurs donnés par les méthodes MCMC. Elle vise à exploiter des informations sur les erreurs dans les évaluations des quantités connues pour réduire l'erreur dans une évaluation d'une quantité

inconnue.
Si notre objectif est d'estimer $\mathbb{E}[X]$. Alors l'estimateur habituel est la moyenne empirique de l'échantillon, défini par,

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{2.9}$$

Cet estimateur est sans biais et convergeant. Supposons que le couple (X_i, U_i), $i = 1, \ldots, n$ est *i.i.d* et que l'espérance $\mathbb{E}[U]$ est connue. La technique des variables de contrôle tire avantage de cette information additionnelle pour but de réduire la variance des estimateurs. Alors pour une valeur θ fixée on définit la variable de contrôle X^* par,

$$X_i^* = X_i - \theta\big(U_i - \mathbb{E}[U]\big)$$

Soit δ_{vc} l'estimateur de $\mathbb{E}[X^*]$, défini par

$$\delta_{vc} = \frac{1}{n}\sum_{i=1}^{n}\Big(X_i - \theta\big(U_i - \mathbb{E}[U]\big)\Big) = \bar{X} - \theta\big(\bar{U} - \mathbb{E}[U]\big). \tag{2.10}$$

On peut vérifier facilement que

$$\begin{aligned}\mathbb{E}[\delta_{vc}] &= \mathbb{E}\big[\bar{X} - \theta(\bar{U} - \mathbb{E}[U])\big] \\ &= \mathbb{E}[\bar{X}] - \theta\big(\mathbb{E}[\bar{U}] - \mathbb{E}\big[\mathbb{E}[U]\big]\big) \\ &= \mathbb{E}[\bar{X}] = \mathbb{E}[X]\end{aligned}$$

Et sa variance est

$$\begin{aligned}\mathbb{V}\mathrm{ar}[\delta_{vc}] &= \mathbb{V}\mathrm{ar}\big[\bar{X} - \theta\big(\bar{U} - \mathbb{E}[U]\big)\big] \\ &= \mathbb{V}\mathrm{ar}[\bar{X}] + \theta^2\mathbb{V}\mathrm{ar}[\bar{U}] - 2\theta\mathbb{C}\mathrm{ov}(\bar{U}, \bar{X})\end{aligned} \tag{2.11}$$

Il existe donc un choix optimal du coefficient θ pour réduire la variance. Si on dérive (2.11) par rapport à θ on aura

$$\theta^* = \frac{\mathbb{C}\mathrm{ov}\big(\bar{X}, \bar{U}\big)}{\mathbb{V}\mathrm{ar}[\bar{U}]}, \tag{2.12}$$

et on remplace θ^* dans (2.11) pour avoir

$$\mathbb{V}\mathrm{ar}[\delta_{vc}] = \big(1 - \sigma_*^2\big)\mathbb{V}\mathrm{ar}[\bar{X}],$$

σ_*^2 étant le coefficient de corrélation entre $\bar{X}$ et $\bar{U}$.
Il est clair que $(1 - \sigma_*^2) \leq 1$, par conséquent

$$\mathbb{V}\mathrm{ar}[\delta_{vc}] \geq \mathbb{V}\mathrm{ar}[\bar{X}]$$

D'où on déduit que sous le choix optimal de θ, utiliser la variable de contrôle résulte en une variance plus petite pour δ_{vc} et que la réduction de variance est d'autant plus forte que les v.a. sont fortement corrélées.

Exemple 2.4. On souhaite estimer,

$$I = E[(X+Y)^{5/4}]$$

où X, Y indépendantes et de même loi appelée *Weibull* de densité

$$f(x) = \frac{3}{2}\sqrt{x}e^{-x^{3/2}}\mathbb{1}_{\mathbb{R}^+}(x).$$

On utilise le fait que : si U est une v.a. uniforme sur $[0,1]$ alors $X = (-LnU)^{2/3}$ suit une loi de Weibull.
Soit $Z = U_1U_2$ une variable de contrôle. Par l'indépendance des variables U_1 et U_2 on déduit

$$\mathbb{E}[Z] = \mathbb{E}[U_1U_2] = \mathbb{E}[U_1]\,\mathbb{E}[U_2] = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$$

D'où l'estimateur basé sur les variables de contrôle est

$$\delta_{vc} = \frac{1}{n}\sum_{i=1}^{n}\Big((X_i+Y_i)^{5/4} - \theta^*\big(Z_i - 1/4\big)\Big).$$

Le code `R` correspondant:

Programme 2.3.

```
n=2000; U=vc=est=numeric(1);
f=function(x) {(-log(x))^(2/3)}
for(i in 1:n){
u=runif(2)
est[i]=(f(u[1])+f(u[2]))^(5/4)
U[i]=u[1]*u[2]
theta=cov(est,U)/var(U)
vc[i]=est[i]-theta*(U[i]-1/4)}
vc[1]=2
m=cumsum(vc)/(1:n)
v=sqrt(cumsum((vc-m)^2))/(1:n)
plot(m,type="l",col=2,xlab="Nombre d'itération",ylim=(c(1.5,2.7)),
lwd=2,ylab="Les moyennes cumulées")
lines(m+2*v,col=2,lwd=1,lty=2)
lines(m-2*v,col=2,lwd=1,lty=2)
mm=cumsum(est)/(1:n)
bor=sqrt(cumsum((est-mm)^2))/(1:n)
```

```
lines(mm,type="l",col=1,lwd=2)
lines(mm+2*bor,col=1,lwd=1,lty=2)
lines(mm-2*bor,col=1,lwd=1,lty=2)
legend(1000,2.7,c("Monte Carlo","variables de contrôle"),
lwd=3, col=c("black","red"))
```

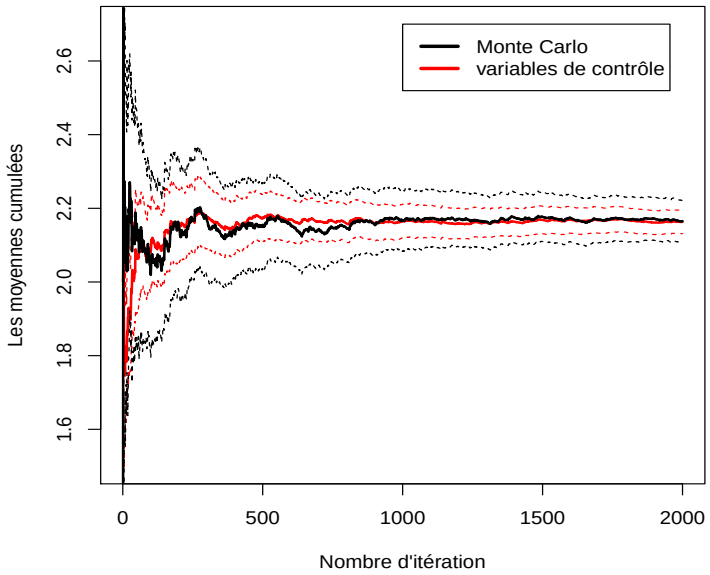

FIG 2.4. Comparaison numérique de la précision des estimations

Dans cet exemple on voit clairement sur la figure [2.4] la différence entre les deux variances des estimations, Monte Carlo en noir et variables de contrôle en rouge.

Pour le tableau des facteurs obtenu est comme suit:

Facteurs de réduction de variance					
	Nombre de simulation				
	$n = 1000$	$n = 10000$	$n = 50000$	$n = 100000$	$n = 200000$
Facteurs	2.58	3.20	3.03	3.12	3.09

Tab 2.4. Facteurs de comparaison entre δ_{vc} et δ_{mc}

D'après ce tableau on voit que la variance de l'estimateur δ_{vc} est diminuée d'environ 3 fois par rapport à la vaviance de l'estimateur δ_{mc} , et que la valeur reste presque

inchangée dans chaque nombre d'itération. On déduit également que la précision et le temps du calcul sont différents de même facteur.

2.5 Méthode de stratification

Etant données une v.a. X et $\{A_i,\, i = 1,\ldots,d\}$ une partition de l'espace d'état de X, appelées *strates* avec $P\big(X \in \bigcup_i A_i\big) = 1$ et $A_i \cap A_j = \emptyset$, $i \neq j$, la quantité $\mathbb{E}[X]$ est approchée par l'estimateur

$$\delta_{St} = \sum_{i=1}^{d} P\big(X \in A_i\big)\mathbb{E}\big[X|X \in A_i\big] = \sum_{i=1}^{d} p_i \mathbb{E}\big[X|X \in A_i\big] \tag{2.13}$$

avec $p_i = P(X \in A_i)$.
Pour prouver que cet estimateur possède une variance réduite, on définit pour $i = 1,\ldots,d$:

$$\bar{X}_{i,n_i} = \frac{1}{n_i}\sum_{j=1}^{n_i} X_j^i, \quad \sigma_i^2 = \mathbb{V}\mathrm{ar}\big(X|X \in A_i\big).$$

On présente alors l'estimateur stratifié par

$$\delta_{St} = \sum_{i=1}^{d} p_i \bar{X}_{i,n_i} \tag{2.14}$$

avec $n = n_1,\ldots,n_d$. On note qu'en raison de l'indépendance (conditionnelle) des sous estimateurs, qu'on a

$$\mathbb{V}\mathrm{ar}(\delta_{St}) = \mathbb{V}\mathrm{ar}\left(\sum_{i=1}^{d} p_i \bar{X}_{i,n_i}\right) = \sum_{i=1}^{d} \frac{p_i^2\sigma_i^2}{n_i}. \tag{2.15}$$

Cependant, pour un choix particulier, les tailles des échantillons pour chaque strate $\{n_i\}$ peuvent être obtenues d'une façon optimale, comme l'indique le théorème suivant.

Théorème 2.2. *(Rubinstein R. Y. and Kroese D. P. 2008)*
Supposons qu'un nombre maximum des échantillons de taille n_i peut être rassemblée, tel que $\sum_{i=1}^{d} n_i = n$, la valeur optimale de n_i, est donnée par

$$n_i^* = n\frac{p_i\sigma_i}{\sum_{j=1}^{d} p_j\sigma_j}. \tag{2.16}$$

Par conséquent, si on remplace (2.16) dans (2.15) on obtient une variance minimale

$$\mathbb{V}\mathrm{ar}(\delta_{St}) = \frac{1}{n}\left[\sum_{i=1}^{d} p_i\sigma_i\right]^2. \tag{2.17}$$

Pour comparer cette variance avec la variance de la moyenne empirique des X_i pour un échantillon de taille n, on calcule la variance de $\bar{X}$;

$$\begin{aligned} n\mathbb{V}\mathrm{ar}(\bar{X}) &= n\mathbb{E}\big[X^2\big] - n\mathbb{E}\big[X\big]^2 \\ &= \sum_{i=1}^{d} p_i \mathbb{E}\big[X^2 | X \in A_i\big] - \left(\sum_{i=1}^{d} p_i \mathbb{E}\big[X^2 | X \in A_i\big]\right)^2. \end{aligned} \tag{2.18}$$

En introduisant les variances conditionnelles et en utilisant deux fois la convexité de la fonction $x \longrightarrow x^2$ et le fait que $\sum_{i=1}^{d} p_i = 1$, on déduit que

$$\begin{aligned} n\mathbb{V}\mathrm{ar}(\bar{X}) &= \sum_{i=1}^{d} p_i \sigma_i^2 + \sum_{i=1}^{d} p_i \mathbb{E}\big[X | X \in A_i\big]^2 - \left(\sum_{i=1}^{d} p_i \mathbb{E}\big[X^2 | X \in A_i\big]\right)^2 \\ &\geq \sum_{i=1}^{d} p_i \sigma_i^2 \geq \left[\sum_{i=1}^{d} p_i \sigma_i\right]^2. \end{aligned} \tag{2.19}$$

On conclut donc que la variance de l'estimateur δ_{St} est inférieure ou égale à celle de $\bar{X}$.

Note importante:

Il faut prendre garde qu'un **mauvais** choix des n_i peut augmenter la variance. Mais on remarque cependant que le choix $n_i = np_i$, s'il n'est pas optimal mais il diminue toujours la variance. En effet, dans ce cas si on remplace n_i dans (2.14) avec inégalité (2.19) on obtient

$$\mathbb{V}\mathrm{ar}(\delta_{St}) = \frac{1}{n}\sum_{i=1}^{d} p_i \sigma_i^2 \geq \mathbb{V}\mathrm{ar}(\bar{X}).$$

Exemple 2.5. Nous utilisons le principe de la méthode de stratification pour calculer l'intégrale suivante

$$\mathcal{I} = \int_0^1 x^2 dx$$

nous choisissons un nombre d'itération $n = 1000$ et un nombre de strates A_i (*intervalles*) $d = 10$ avec la même distance.
Donc nous avons $p_i = P(X \in A_i) = \frac{1}{d} = \frac{1}{10}$ pour $i = 1, \ldots, d$.
Le nombre d'itération dans chaque strate $N_i = Np_i = 1000/10 = 100$, $\forall i$.

L'implémentation en `R` est comme suit:

Programme 2.4.

```
n=1000; d=10; Ni=100; p=1/d;
f=function(x){x^2}
s=q=U=mu=x=u=numeric(1);
for(h in 1:n){
for(j in 1:d){m=0;
for(i in 1:Ni){
u[i]=runif(1)
c=x[i]=f(u[i])
U[(j-1)*d+i]=c
m=m+x[i]}
q[j]=mean(x)}
s[h]=sum(q)*p
M[h]=mean(U)}
mx=cumsum(s)/1:n
vx=sqrt(cumsum((s-mx)^2))/1:n
plot(mx,type="l",lwd=2,xlab="Nombre d'itération",
ylab="Les moyennes cumulées")
lines(mx+2*vx,col=1,lwd=1,lty=2)
lines(mx-2*vx,col=1,lwd=1,lty=2)
for(i in 1:n){
xu=f(runif(Ni*d))
M[i]=mean(xu)}
m=cumsum(M)/1:n
v=sqrt(cumsum((M-m)^2))/1:n
lines(m,type="l",col=2,lwd=2)
lines(m+2*v,col=2,lwd=1,lty=2)
lines(m-2*v,col=2,lwd=1,lty=2)
```

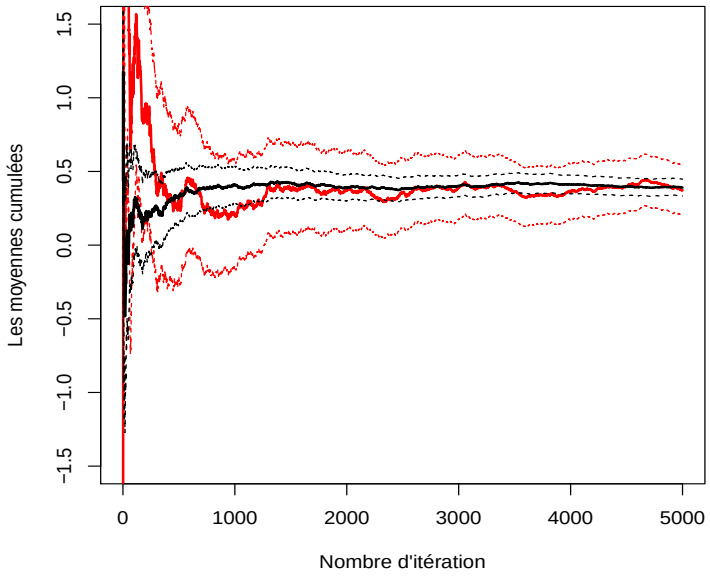

FIG 2.5. Comparaison numérique de la précision des deux estimations $\bar{X}$ et $\bar{X}_{st,N}$

La figure [2.5] montre la différence entre la variance de l'estimateur $\bar{X}$ en rouge et celle de δ_{St} en noir.
Le tableau correspondant est comme suit

Facteurs de réduction de variance					
	Nombre de simulation				
Estimateur	$n = 1000$	$n = 10000$	$n = 50000$	$n = 100000$	$n = 200000$
Facteurs	2.56	3.3	3.23	4.21	5.10

Tab 2.5. Facteurs de comparaison entre δ_{St} et δ

On remarque à travers le tableau que le facteur augmente avec le nombre de simulation : environ 3 fois plus grand pour $n = 1000$ et 5 fois pour $n = 200000$.

2.6 Approximation Riemannienne

Les techniques de simulation et, en particulier, les méthodes MCMC visent souvent à rapprocher des intégrales de la forme

$$\mathbb{E}[h(X)] = \int_{\mathcal{X}} h(x)f(x)dx \tag{2.20}$$

pour les fonctions d'intérêt intégrables h. Une fois qu'un échantillon $(X_1,\ldots,X_n)$ est produit suivant f, l'estimateur standard est toujours la moyenne empirique donnée par

$$\delta_T = \frac{1}{T}\sum_{i=1}^{T} h(X_i). \tag{2.21}$$

A fin d'améliorer la performance des estimateurs, Anne Philippe et C.P. Robert [1] proposent une autre approximation appelée somme de Riemann définie par

$$\delta_T^R = \sum_{i=1}^{T-1} \Big(X^{[i+1]} - X^{[i]}\Big) f\big(X^{[i]}\big) h\big(X^{[i]}\big), \tag{2.22}$$

avec $X^{[1]} \leq , \ldots , \leq X^{[T]}$ est la statistique d'ordre déduite de cet échantillon.
D'après Philippe cet estimateur converge bien vers $\mathbb{E}_f[h(X)]$.
Comme dans la plupart des méthodes MCMC, f est souvent connue à un facteur multiplicatif prés *i.e.* $f(x) \propto \tilde{f}(x)$, la représentation alternative de (2.22) sera donc

$$\delta_T^R = \frac{\displaystyle\sum_{i=1}^{T-1} \Big(X^{[i+1]} - X^{[i]}\Big) \tilde{f}\big(X^{[i]}\big) h\big(X^{[i]}\big)}{\displaystyle\sum_{i=1}^{T-1} \Big(X^{[i+1]} - X^{[i]}\Big) \tilde{f}\big(X^{[i]}\big)} \tag{2.23}$$

Il est bien de noter qu'une relation de récurrence existe entre δ_T^R et δ_{T-1}^R qui sera utile dans implémentation, qui sert, en fait, pour éviter beaucoup de calculs en plus,

$$\delta_T^R = \delta_{T-1}^R + \Big(X^{[i+1]} - X^{[i]}\Big)\Big[h\big(X^{[i]}\big) f\big(X^{[i]}\big) - h\big(X^{[i-1]}\big) f\big(X^{[i-1]}\big)\Big].$$

Exemple 2.6. *(Philippe A. and Robert C. P. 2000)*
Considérons la densité

$$f_0(x) \propto \frac{e^{-x^2/2}}{\big(1+(x-x_0)^{\nu}\big)},$$

qui peut être représentée comme une densité marginale de

$$g(x,y) \propto e^{-x^2/2} y^{\nu-1} e^{-(1+(x-x_0)^2 y/2},$$

1. Disponible sur: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.1604&rep=rep1&type=pdf

avec les distributions conditionnelles suivantes

$$X \mid y \sim \mathcal{N}\Big(x_0 y/(1+y),\, 1/(1+y)\Big)$$

$$Y \mid x \sim \mathcal{G}a\Big(\nu,\, \big(1+(x-x_0)^2\big)/2\Big).$$

L'échantillonneur de Gibbs correspondant peut être donc appliqué dans ce cas. On initialise les valeurs $x_0 = 0$ et $\nu = 2$, l'implémentation en R est comme suit:

Programme 2.5.

```
N=2000;  nu=2;  a=0;
y=x=mv=bv=av=numeric(1);
f=function(x,a=0,nu=2) {(exp(-(x^2)/2))/((1+(x-a)^2)^nu)}
y[1]=.3;
x[1]=rnorm(1,a,1/(1+y[1]))
for(i in 2:N){
y[i]=rgamma(1,nu,(1+(x[i-1]-a)^2)/2)
x[i]=rnorm(1,a*y[i]/(1+y[i]),sqrt(1/(1+y[i])))
v=sort(x)  # pour ordonner le vecteur
bv=v[-1]   # suppression de la première composante
av=v[-i]   # suppression de la ième composante
mv[i]=sum((bv-av)*f(av)*av)/sum((bv-av)*f(av)) }
mm=cumsum(mv)/1:N
va=sqrt(cumsum((mv-mm)^2))/1:N
plot(mv,type="l",col=2,ylim=c(-0.2,0.2),lwd=2,
xlab="Nombre d'itération",ylab="Les moyennes cumulées")
lines(mv+2*va,lwd=1,col=2,lty=2)
lines(mv-2*va,lwd=1,col=2,lty=2)
abline(a=0,b=0,col=3,lty=2)
m=cumsum(x)/1:N
var=sqrt(cumsum((x-m)^2))/1:N
lines(m,type="l",col=1,lwd=2)
lines(m+2*var,col=1,lwd=1,lty=2)
lines(m-2*var,col=1,lwd=1,lty=2)
```

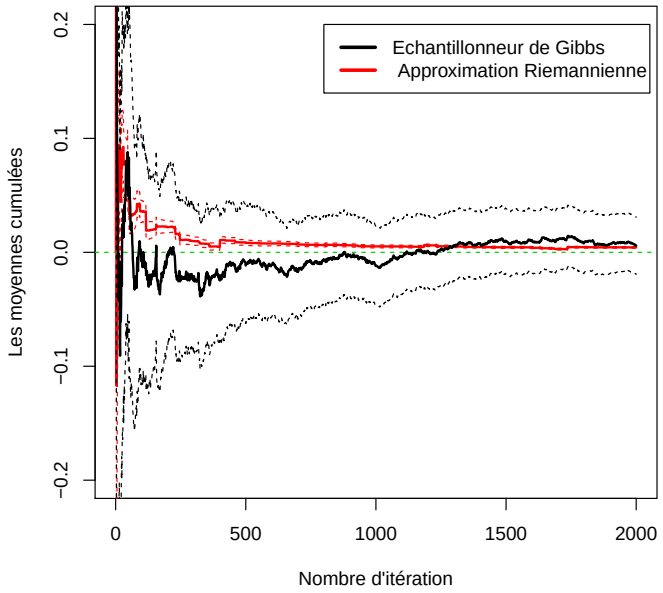

FIG 2.6. Comparaison numérique de la précision des deux estimations δ_T et δ_T^R

La figure [2.6] représente une comparaison entre l'évolution de l'estimateur δ_T avec l'intervalle de confiance à 95% en noir, qui possède, en fait, une variance plus grande par rapport à l'approximation par sommes de Riemann δ_T^R présenté en rouge.

Le tableau correspondant

Facteurs de réduction de variance					
	Nombre de simulation				
	$n = 1000$	$n = 10000$	$n = 50000$	$n = 100000$	$n = 200000$
Facteurs	14.25	171.40	247.45	333.60	1512.52

Tab 2.6. Facteurs de comparaison entre δ_T et δ_T^R

On déduit à travers le tableau que l'estimateur par l'approximation Riemannienne réduit la variance de façon remarquable et qu'elle décroît rapidement en fonction de nombre simulation par rapport à la variance de l'estimateur δ_T.

Chapitre 3

L'usage des Variables de Contrôle

3.1 Introduction

Les méthodes MCMC se révèlent souvent être les seules méthodes pratiques pour approximer les quantités extrêmement complexes, pour cela le domaine de recherche correspondant se situe aux confluents de nombreuses disciplines, telles les statistiques, les processus stochastiques, l'inférence bayésienne, l'informatique, ainsi que les différentes sciences appliquées, qui les adoptent comme outil indispensable pour pouvoir faire des calculs plus exacte.
Dans le chapitre précédent, nous avons vu que plusieurs méthodes ont été proposées, qui visent à réduire la variance à fin d'accélérer la convergence vers la valeur à estimer pour avoir justement une meilleure précision, bien que ces méthodes portent des améliorations considérables, mais chacune d'elle a ces limites, pour cela la recherche ne cesse pas de donner d'autre méthodes tout en essayant soit de proposer d'autre alternatifs ou d'améliorer les méthodes classiques. D'ailleurs, une autre approche tout à fait récente est proposée par Petros Dellaportas Ioannis Kontoyiannis (2012), qui s'appuie sur l'idée des variables de contrôle a réussi à résoudre beaucoup de problèmes liés à cette problématique.

3.2 Mise en œuvre

Soit $\{X_n\}_{n\geq 0}$ une chaîne de Markov à temps discret qui prend ses valeurs dans un espace dénombrable $\mathbb{E}$, muni d'une tribu $\mathcal{B}$ avec un état initial $X_0 = x$, $P(x,dy)$ son noyau de transition défini par:

$$P(x,dy) = P\big[X_{n+1} \in A \mid X_n = x\big], \quad x \in \mathbb{E}, \;\; A \in \mathcal{B}$$

Etant donnée une fonction $F : \mathbb{E} \longrightarrow \mathbb{R}$. Comme dans toutes les applications, notamment dans les méthodes MCMC, on intéresse toujours à calculer l'espérance $\pi(F) = E_\pi(F) = \int F d\pi$. Bien que le calcul direct de cette espérance n'est pas généralement une question simple même parfois impossible, Mais c'est possible de construire une chaîne de Markov $\{X_n\}$ facile à simuler qui possède π comme l'unique mesure invariante. Sous quelques conditions, la distribution de X_n converge vers π qui peut être rendue précise généralement.
Commençons alors par quelques résultats et définitions :

$$PF(x) = E_x[F(X_1)] = E\big[F(X_1) \mid X_0 = x)\big]$$

Alors pour tout état initial x,

$$P^n F(x) = E\big[F(X_n) \mid X_0 = x)\big] \longrightarrow \pi(F) \quad pour \quad n \longrightarrow \infty$$

Pour une classe appropriée des fonctions $F : \mathbb{E} \longrightarrow \mathbb{R}$, le taux de convergence peut être écrit sous la forme suivante:

$$\hat{F}(x) = \sum_{n=0}^{\infty} \Big[P^n F(x) - \pi(F)\Big], \tag{3.1}$$

dans ce cas, $\hat{F}$ satisfait *l'équation de Poisson* qui s'écrit sous la forme

$$P\hat{F} - \hat{F} = -F + \pi(F). \tag{3.2}$$

Mais la solution de *l'équation de Poisson*, même pour les fonctions simples, est une tâche non triviale, et notamment dans notre cas, la solution est impossible (voir, par exemple, les commentaires appropriés de Henderson (1997)[15] et Meyn (2007)[20]).
Les résultats ci-dessus décrivent comment la distribution de X_n converge vers π.
En termes d'estimation, les quantités d'intérêt sont les moyennes ergodiques,

$$\mu_n(F) = \frac{1}{n} \sum_{i=0}^{n-1} F(X_i). \tag{3.3}$$

Encore, sous quelques conditions appropriées, le théorème ergodique assure que,

$$\mu_n(F) \longrightarrow \pi(F) \quad pour \quad n \longrightarrow \infty. \tag{3.4}$$

Pour notre cas, le taux de convergence est mesuré par le théorème de la Limite Centrale associé,

$$\sqrt{n}\big[\mu_n(F)-\pi(F)\big]=\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}\big[F(X_i)-\pi(F)\big]\xrightarrow{L}N(0,\sigma_F^2)\quad pour\quad n\longrightarrow\infty$$

Ici, σ_F^2 est la variance asymptotique de F , donnée par

$$\sigma_F^2=\lim_{n\to\infty}Var_\pi\big(\sqrt{n}\mu_n(F)\big),$$

qu'on peut également l'exprimer avec les $\hat{F}$:

$$\sigma_F^2=\pi\Big(\hat{F}^2-(P\hat{F})^2\Big) \tag{3.5}$$

Les résultats dans les équations (3.1) et (3.5) indiquent clairement qu'il est utile de connaître la solution $\hat{F}$ de *l'équation de Poisson* pour F. Mais en général c'est une tâche non triviale, alors que la fonction $\hat{F}$ joue un rôle central dans tous les développements qui viennent.

3.3 Les variables de contrôle

Soit une chaîne de Markov $\{X_n\}$ avec un noyau de transition P et de mesure invariante π, la moyenne ergodique $\mu_n(F)$ est utilisée pour estimer la moyenne $\pi(F)=\int Fd\pi$, bien que les estimateurs $\mu_n(F)$ convergent vers $\pi(F)$ quand n tend vers l'infini, mais le problème majeur qu'on rencontre souvent dans la pratique c'est que la variance asymptotique σ_F^2 associée est grande, par conséquent, la convergence vers $\pi(F)$ sera très lente. Afin de réduire cette variance, on emploie l'idée des variables de contrôle.
Soit alors $U=(U_1,U_2,\ldots,U_k)^t$ un vecteur des variables de contrôle tel que $U_i:\mathbb{E}\longrightarrow\mathbb{R}$ avec $E(U_i)=0$, $\forall i=1,2,\ldots,k$; soit aussi $\theta=(\theta_1,\theta_2,\ldots,\theta_k)^t$ vecteur des constantes dans $\mathbb{R}^k$.
On définit donc:

$$F_\theta=F-\big\langle\theta,U\big\rangle=F-\sum_{i=1}^{k}\theta_iU_i \tag{3.6}$$

On considère les estimateurs modifiés pour $\pi(F)$,

$$\mu_n(F_\theta)=\mu_n(F)-\langle\theta,\mu_n(U_i)\rangle=\mu_n(F)-\sum_{i=1}^{k}\theta_i\mu_n(U_i) \tag{3.7}$$

Le théorème ergodique garantit que les estimateurs $\mu_n(F_\theta)$ convergent avec probabilité un, et on remarque aussi qu'avec un choix particulier de U et θ, on peut réduire significativement la variance asymptotique σ_F^2 de ces estimateurs modifiés de $\mu_n(F)$.

Pour voir l'importance de cette méthode on renvoi le lecteur à la section (2.4).
Maintenant et dans la suite, on va baser sur le choix de ces variables proposées par Henderson (1997) [15].
Pour des fonctions (π-intégrables) arbitraires: $G_i : E \longrightarrow \mathbb{R}$.
On définit:

$$U_i = G_i - PG_i, \qquad i = 1,2,\ldots,k$$

D'ailleurs, l'invariance du π sous P et l'intégrabilité de G_i garantissent que $E(U_i) = 0$.
Dans la suite on donne quelques orientations simples pour choisir des fonctions $\{G_i\}$ pour que les $\{U_i\}$ soient effectivement des variables de contrôle.
En premier lieu, on suppose qu'on est libre de choisir les fonctions $\{G_i\}$. Sans perte de généralité, on pose $k = 1$ et $\theta = 1$, alors, le but est de rendre la variance asymptotique de $F - U = F - G + PG$ comme la plus petite variance possible, mais en vue de *l'équation de Poisson*, on remarque que le choix $G = \hat{F}$ conduit à:

$$F - U = F - \hat{F} + P\hat{F} = \pi(F)$$

Ce qui donne une variance égale à zéro. Par conséquent, le principe de sélection de G est:
choisir la variable de contrôle $U = G - PG$ ***avec*** $G \simeq \hat{F}$**.**
Comme c'est mentionné ci-dessus, il est en général impossible de calculer $\hat{F}$ pour les modèles réalistes utilisés dans les applications. Mais il est souvent possible de proposer une fonction G qui rapproche $\hat{F}$. Pour cela Dellaportas et Kontoyiannis (2012)[7] proposent de rapprocher $\hat{F}$ par des combinaisons linéaires de $\{G_i\}$, *i.e.* $\hat{F} \simeq \sum_{i=1}^{k} G_i$.
Une fois que les fonctions $\{G_i\}$ sont sélectionnées, on obtient donc les estimateurs modifiés $\mu_n(F_\theta)$,

$$F_\theta = F - \langle \theta,U \rangle = F - \langle \theta,G \rangle + \langle \theta,PG \rangle.$$

avec $PG = (PG_1,PG_2,\ldots,PG_k)^t$.
La question suivante est de choisir θ de sorte que la variance,

$$\sigma_\theta^2 = \sigma_{F_\theta}^2 = \pi\Big(\hat{F}_\theta^2 - (P\hat{F}_\theta)^2\Big)$$

soit minimale. On note que $\hat{U}_i = G_i \quad et \quad \hat{F}_\theta = \hat{F} - \langle \theta,G \rangle$.
Par conséquent, on peut développer σ_θ^2 en utilisant la formule (3.5) et la linéarité pour obtenir:

$$\sigma_\theta^2 = \sigma_F^2 - 2\pi\Big(\hat{F}\langle \theta,G \rangle - P\hat{F}\langle \theta,PG \rangle\Big) + \pi\Big(\langle \theta,G \rangle^2 - \langle \theta,PG \rangle^2\Big). \tag{3.8}$$

Pour trouver donc le vecteur θ^* optimal, on minimise σ^2_θ. Pour se faire alors on dérive σ^2_θ par rapport à θ pour avoir

$$\Gamma(G)\theta^* = \pi\Big(\hat{F}G - (P\hat{F})(PG)\Big),$$

avec $\Gamma(G) = \Gamma(G)_{ij} = \pi\Big(G_iG_j - (PG_i)(PG_j)\Big)$

par conséquent, si $\Gamma(G)$ inversible alors,

$$\theta^* = \Gamma(G)^{-1}\pi\Big(\hat{F}G - (P\hat{F})(PG)\Big). \tag{3.9}$$

On remarque malheureusement que θ^* dépend de $\hat{F}$ donc le problème est toujours resté[1]. Mais au moins, on peut interpréter le résultat obtenu. Pour simplifier les choses, on considère le cas simple $U = G - PG$, donc la valeur de θ^* dans (3.9) peut se simplifier par,

$$\theta^* = \frac{\pi\big(\hat{F}G - (P\hat{F})(PG)\big)}{\pi(G^2 - PG^2)} = \frac{\pi\big(\hat{F}G - (P\hat{F})(PG)\big)}{\sigma^2_U}. \tag{3.10}$$

Alternativement, si on développe l'expression, $\sigma^2_\theta = \lim_{n\to\infty} Var_\pi\big(\sqrt{n}\mu_n(F_\theta)\big)$, on aura,

$$\sigma^2_\theta = \sigma^2_F + \theta^2\sigma^2_U - 2\theta \sum_{n=-\infty}^{\infty} Cov_\pi\big(F(X_0),U(X_n)\big),$$

de sorte que le vecteur θ^* puisse être exprimé également sous forme,

$$\theta^* = \frac{1}{\sigma^2_U} \sum_{n=-\infty}^{\infty} Cov_\pi\big(F(X_0),U(X_n)\big). \tag{3.11}$$

Ici Cov_π est la covariance pour la version stationnaire de la chaîne, *i.e*, depuis que $\pi(U) = 0$, $Cov_\pi\big(F(X_0),U(X_n)\big) = E_\pi\big[F(X_0),U(X_n)\big]$, avec $X_0 \sim \pi$, alors (3.11) donne une variance asymptotique optimale,

$$\sigma^2_{\theta^*} = \sigma^2_F - \frac{1}{\sigma^2_U}\Big[\sum_{n=-\infty}^{\infty} Cov_\pi\big(F(X_0),U(X_n)\big)\Big]^2.$$

Par conséquent, afin de réduire la variance, il est souhaitable que la corrélation entre F et U soit la plus grande possible. Ceci mène au deuxième principe pour choisir des fonctions de base :

Choisir les variables de contrôle $U = G - PG$ de sorte que chaque U_i soit fortement corrélée avec le F.

D'ailleurs, si on compare les expressions (3.10) et (3.11) on peut déduire que,

$$\sum_{n=-\infty}^{\infty} Cov_\pi\Big(F(X_0),U(X_n)\Big) = \pi\Big(\hat{F}G - (P\hat{F})(PG)\Big) \tag{3.12}$$

1. Ce problème sera reposé dans la section (1.5).

3.4 Estimation du vecteur des coefficients optimaux θ^*

Rappelant qu'une fois les fonctions de base $\{G_i\}$ ont été choisis, on aura donc le vecteur des coefficients optimaux écrit:

$$\theta^* = \Gamma(G)^{-1}\pi\Big(\hat{F}G - (P\hat{F})(PG)\Big).$$

On peut également écrire $\Gamma(G)$ autrement:

$$\begin{aligned}\Gamma(G)_{ij} &= \pi\Big(G_iG_j - (PG_i)(PG_j)\Big)\\ &= \pi\Big(\hat{U}_iG_j - (P\hat{U}_i)(PG_j)\Big)\\ &= \sum_{n=-\infty}^{\infty} Cov_\pi\Big(U_i(X_0),G_j(X_n)\Big).\end{aligned}$$

Ceci indique que $\Gamma(G)$ a la structure d'une matrice de covariance, et en particulier, elle suggère que la matrice $\Gamma(G)$ devrait être semi-définie positive.

Proposition 3.1. *[7] Soit $K(G)$ dénoter la matrice de covariance des variables aléatoires*

$$Y_i = G_i(X_1) - PG_i(X_0), \qquad j = 1,2,\ldots,k$$

Ici $X_0 \sim \pi$ alors $\Gamma(G) = K(G)$ pour tous $1 \leq i,j \leq k$,

$$\pi\Big[G_iG_j - (PG_i)(PG_j)\Big] = K(G)_{ij} = E_\pi\Big[\Big(G_i(X_1) - PG_i(X_0)\Big)\Big(G_j(X_1) - PG_j(X_0)\Big)\Big]. \tag{3.13}$$

Démonstration 3.1. On peut exploiter le côté droit de (3.13) pour avoir

$$\pi(G_iG_j) - E_\pi\big[G_i(X_1)PG_j(X_0)\big] - E_\pi\big[G_j(X_1)PG_i(X_0)\big] + \pi\Big((PG_i)(PG_j)\Big),$$

On remarque que le deuxième et le troisième terme ci-dessus sont les deux égaux au quatrième, de plus le deuxième terme peut être récrit sous la forme

$$E_\pi\Big\{E\Big[G_i(X_1)PG_j(X_0) \mid X_0\Big]\Big\} = E_\pi\Big\{E\Big[G_i(X_1) \mid X_0\Big]PG_j(X_0)\Big\} = \pi\Big((PG_i)(PG_j)\Big),$$

même chose pour le troisième terme.

□

Par conséquent, en utilisant la proposition (3.1), le vecteur θ^* peut s'écrire:

$$\theta^* = K(G)^{-1}\pi\Big(\hat{F}G - (P\hat{F})(PG)\Big). \tag{3.14}$$

Maintenant on suppose que la chaîne $\{X_n\}$ est réversible, soit alors $\Delta = P - I$ un générateur de $\{X_n\}$, la réversibilité implique que Δ est un operateur auto adjoint

linéaire dans l'espace $L_2(\pi)$.
En autre terme, $\pi(F\Delta G) = \pi(\Delta FG)$ pour les fonctions F,$G \in L_2(\pi)$.

Proposition 3.2. *[7] Si la chaîne $\{X_n\}$ est réversible, alors le vecteur θ^* pour les variables de contrôle $U_i = G_i - PG_i$, $i = 1,2,\ldots,k$, s'écrit plus tôt:*

$$\theta^* = \theta^*_{rev} = \Gamma(G)^{-1}\pi\Big((F - \pi(F))(G + PG)\Big). \tag{3.15}$$

Ou

$$\theta^*_{rev} = K(G)^{-1}\pi\Big((F - \pi(F))(G + PG)\Big). \tag{3.16}$$

Démonstration 3.2. Soit $\bar{F} = F - \pi(F)$ dénoter la version centrée de F, on rappelle que $\hat{F}$ résout l'équation de Poisson pour F, ainsi $P\hat{F} = \hat{F} - \bar{F}$. Par conséquent, employant le fait que Δ est auto adjoint sur chaque composant de G,

$$\begin{aligned}
\pi\Big(\hat{F}G - (P\hat{F})(PG)\Big) &= \pi\Big(\hat{F}G - (\hat{F} - \bar{F})(PG)\Big) \\
&= \pi\Big(\bar{F}PG - \hat{F}\Delta G\Big) \\
&= \pi\Big(\bar{F}PG - \Delta\hat{F}G)\Big) \\
&= \pi\Big(\bar{F}PG - \bar{F}G\Big) \\
&= \pi\Big(\bar{F}(G + PG)\Big).
\end{aligned}$$

Combinant ceci avec (3.9) et (3.14), respectivement, prouve les deux réclamations de la proposition.

□

L'expression (3.16) suggère l'estimation de θ^* par:

$$\hat{\theta}_{n,K} = K_n(G)^{-1}\Big[\mu_n(F(G + PG)) - \mu_n(F)\mu_n(G + PG)\Big], \tag{3.17}$$

avec

$$(K_n(G))_{ij} = \frac{1}{n-1}\sum_{t=1}^{n-1}\Big(G_i(X_t) - PG_i(X_{t-1})\Big)\Big(G_j(X_t) - PG_j(X_{t-1})\Big).$$

D'ou estimateur $\mu_n(F_{\hat{\theta}_{n,K}})$ pour $\pi(F)$ basé sur les variables de contrôle $U = G - PG$ et les coefficients $\hat{\theta}_{n,K}$ estimés est défini comme suit:

$$\mu_{n,K}(F) = \mu_n(F_{\hat{\theta}_{n,K}}) = \mu_n(F) - \big\langle\hat{\theta}_{n,K},\mu_n(U)\big\rangle. \tag{3.18}$$

3.5 Le choix des fonctions de base

Soit $\{X_n\}$ une chaîne de Markov ergodique et réversible avec π sa mesure invariante, et soit $\pi(F)$ la moyenne à estimer par l'échantillon provenant de cette chaîne. D'après les sections précédentes, l'objectif principal est de choisir les fonctions de base G_i qui devraient approximer effectivement la solution $\hat{F}$ de *l'équation de Poisson* pour F comme combinaisons linéaires des G_i, *i.e.* $\hat{F} \simeq \sum_{i=1}^{k} \theta_i G_i$. Dans le cas d'un échantillonneur aléatoire de Gibbs avec une densité cible gaussienne, *l'équation de Poisson* peut être résolue explicitement, et sa solution est d'une forme particulièrement simple.

Théorème 3.1. *[7] Soit $\{X_n\}$ une chaîne de Markov construite par l'échantillonneur de Gibbs utilisée pour simuler une loi de Gauss multivariée $\pi \sim N(\mu,\Sigma)$ dans $\mathbb{R}^k$. Si l'objectif est d'estimer la moyenne pour la première composante de π, alors, soit $F(x) = x^{(1)}$ pour $x = (x^{(1)},x^{(2)},\ldots,x^{(k)})^t \in \mathbb{R}^k$, la solution $\hat{F}$ de l'équation de Poisson Pour F peut être exprimée par des combinaisons linéaires des fonctions de base $G_i(x) = x^{(i)}$, $x \in \mathbb{R}^k$,$1 \leq i \leq k$*

$$\hat{F} = \sum_{i=1}^{k} \theta_i G_i. \tag{3.19}$$

D'ailleurs, en écrivant $Q = \Sigma^{-1}$, le vecteur de coefficient θ dans l'équation (3.19) est donné par la première rangée de la matrice $k(I - A)^{-1}$ avec $A_{ij} = -Q_{ij}/Q_{ii}$, $1 \leq i \neq j \leq k$, $\forall i$ $A_{ii} = 0$, et $I - A$ est toujours inversible.

Démonstration 3.3. (Voir Dellaportas P. et Kontoyiannis I. 2012[2]).

Supposons que des échantillons proviennent d'une distribution gaussienne multivariée sont simulés par échantillonneur de Gibbs. Si on veut estimer la moyenne de l'une de ses composantes, alors d'après le théorème (3.1), on peut utiliser les fonctions de base $G_i(x) = x^{(i)}$ pour construire le vecteur des variables de contrôle $U = G - PG$, par conséquent, la variance de l'estimateur,

$$\mu_{n,K}(F) = \mu_n(F_{\hat{\theta}_{n,K}}) = \mu_n(F) - \left\langle \hat{\theta}_{n,K},\mu_n(U) \right\rangle,$$

sera remarquablement petite par rapport à la moyenne ergodique $\mu_n(F)$.

2. Disponible sur: http://www.cs.aueb.gr/ yiannisk/PAPERS/cvmcmcJ.pdf

Méthodologie de base

1. Données:
   - Une distribution multivariée a posteriori $\pi(x) = \pi(x^{(1)},x^{(2)},\ldots,x^{(d)})$.
   - Une chaîne de Markov $\{X_n\}$ réversible avec loi stationnaire π.
   - Echantillon de taille N provenant de la chaîne $\{X_n\}$.
2. Le But:
   - Estimer la moyenne a posteriori μ^i de x^i.
3. Définitions:
   - $F(x) = x^{(i)}$.
   - Fonctions de base $G_i(x) = x^{(i)}$ et $PG_i(x) = E[X_{n+1}^{(i)}|X_n = x]$ peut être calculée explicitement.
   - Les variables de contrôle correspondantes $U_i = G_i - PG_i$.
4. Estimations:
   - Le vecteur optimal θ^* par $\hat{\theta}_{n,K}$ défini dans (3.17).
   - La quantité μ^i par l'estimateur modifié $\mu_{n,K}(F)$ donnée par (3.18).

3.6 Application

Exemple 3.1. Soit $(X,Y) \sim \pi(x,y)$ une distribution normale bivariée, ici, sans perte de généralité, on prend $\mathbb{E}(X) = \mathbb{E}(Y) = 0$, et on pose $\mathbb{V}\text{ar}(X) = 1$, $\mathbb{V}\text{ar}(Y) = \tau^2$ et $\mathbb{C}\text{ov}(X,Y) = \rho\tau$ pour $\rho \in (-1,1)$, on donne des valeurs initiales arbitraires $x_0 = x$ et $y_0 = y$. Les lois conditionnelles sont définies par:

$$Y|X = x \sim \mathcal{N}\big(\rho\tau x, \tau^2(1-\rho^2)\big) \quad \text{et} \quad X|Y = y \sim \mathcal{N}\big(\frac{\rho}{\tau}y, 1-\rho^2\big)$$

Pour estimer l'espérance de X sous π on pose $F(x,y) = x$, et on définit les fonctions de base $G_1(x,y) = x$ et $G_2(x,y) = y$. Les fonctions correspondantes PG_1 et PG_2 sont faciles à calculer

$$PG_1 = \frac{1}{2}\Big[x + \frac{\rho y}{\tau}\Big] \quad \text{et} \quad PG_2 = \frac{1}{2}\Big[y + \rho\tau x\Big].$$

Pour estimer $\mathbb{E}[X]$ par $\mu_n(F)$ et $\mu_{n,K}(F)$, On fixe les valeurs des paramètres en $\rho = 0.99$ et $\tau^2 = 10$, de sorte que les deux composantes soient fortement corrélées comme la montre la figure suivante.

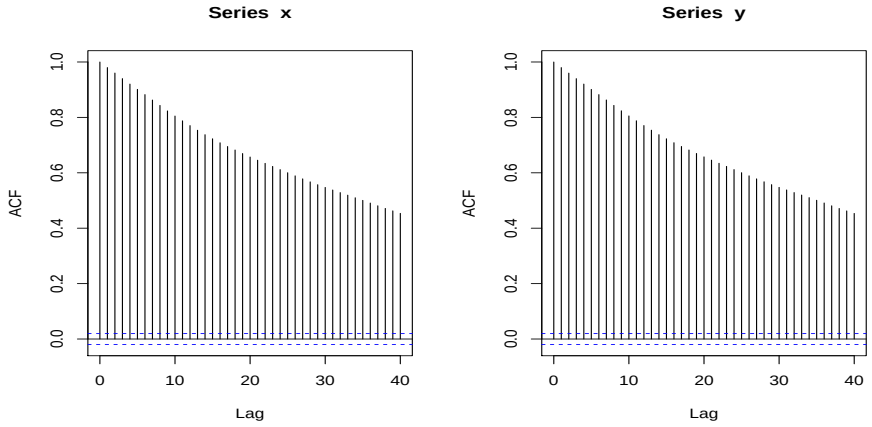

FIG 3.1. Histogrammes d'autocorrélation pour les échantillons X et Y

Sur la figure [3.2] on remarque clairement l'effet d'autocorrélation sur l'évaluation des deux chaînes.

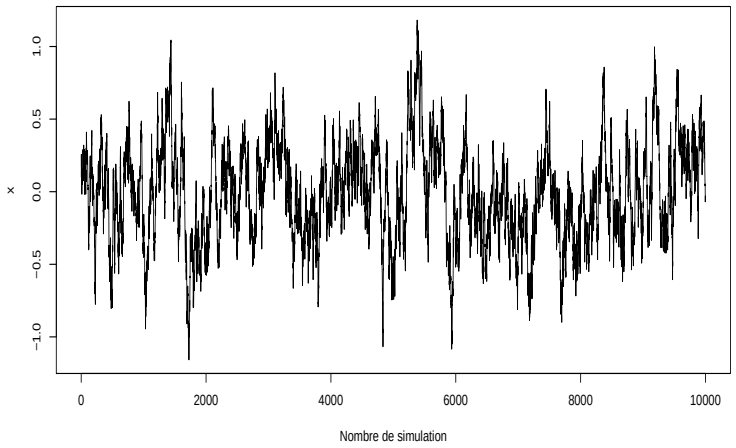

FIG 3.2. Evolution de la chaîne X_n

D'après ce que nous avons détaillé dans la section (1.8) (Problèmes de convergence) ces deux figures nous indiquent que le déplacement de la chaîne sur le support de X est très lent, et par conséquent, la convergence vers la loi stationnaire va être aussi très lente, comme on le voit sur la figure [3.3] où l'évolution de l'estimateur $\mu_n(F)$ apparaît instable même après 20 milles itérations.

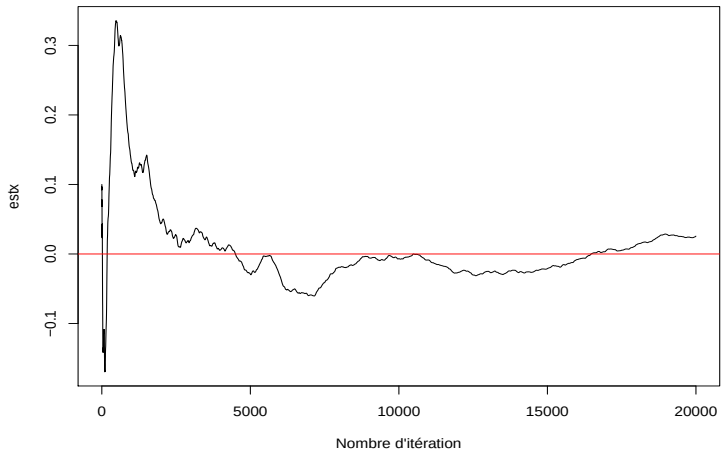

FIG 3.3. Evolution des moyennes cumulées de la chaîne X_n

Nous utilisons l'échantillonneur de Gibbs pour produire un échantillon distribué sous la loi π puis on examine l'exécution de l'estimateur modifié $\mu_{n,K}(F)$, pour le compare avec l'exécution les moyennes ergodiques standard $\mu_n(F)$, le programme R correspondant est comme le suit:

Programme 3.1.

```
n=20000; a=0.99; T=sqrt(10);
G=function(i,x){if(i==1) x[1] else x[2]}
f=function(x){x[1]}
PG=function(i,x){if(i==1){(1/2)*(x[1]+(a*x[2])/T)}
else {(1/2)*(x[2]+a*x[1]*T)}}
x=numeric(1);
uKF=y=mm=uf=x; mm=5;
uKF[1]=mof=uf[1]=x[1]=0.1;
y[1]=rnorm(1,a*T*x[1],sqrt((T^2)*(1-(a^2))))
kl=k=matrix(rep(0,4),nrow=2)
z=c(x[1],y[1]);
a1=c(f(G(1,z)+PG(1,z)),f(G(2,z)+ PG(2,z)))
a2=c(G(1,z)+PG(1,z),G(2,z)+ PG(2,z))
u=c(G(1,z)-PG(1,z),G(2,z)-PG(2,z))
for(i in 2:n){ q=z;
x[i]=rnorm(1,(a/T)*y[i-1],sqrt(1-(a^2)))
y[i]=rnorm(1,a*T*x[i],sqrt((T^2)*(1-(a^2))))
z=c(x[i],y[i]);
a1=a1+c(f(G(1,z)+PG(1,z)),f(G(2,z)+PG(2,z)))
aa1=a1/i;
a2=a2+c(G(1,z)+PG(1,z),G(2,z)+PG(2,z))
aa2=aa1=a2/i;
mof=mof+f(z);
uf[i]=mof/i; ak=k;
for(j in 1:2){
for(h in 1:2){
k[j,h]=k[j,h]+(G(j,z)-PG(j,q))*(G(h,z)-PG(h,q))}}
KG=ak/(i-1);
u=u+c(G(1,z)-PG(1,z),G(2,z)-PG(2,z))
U=u/i;
if (i>3){A=solve(KG)%*%(aa1-uf[i]*aa2);
uKF[i]=(uf[i]-(A[1]*U[1]+A[2]*U[2]))/mm}
if(i%%250==2){
```

```
v[i]=uKF[i]}
else {v[i]=NA}}
plot(uf,type="l",ylab="Les moyennes cumulées",xlab=" ")
lines(v,col="blue",type="p",lwd=1)
abline(a=0,b=0,col="red")
```

La figure [3.4] montre effectivement que la variance de l'estimateur $\mu_{n,K}(F)$ en bleu possède une variance réduite par rapport à l'estimateur $\mu_n(F)$ tracé en noir.

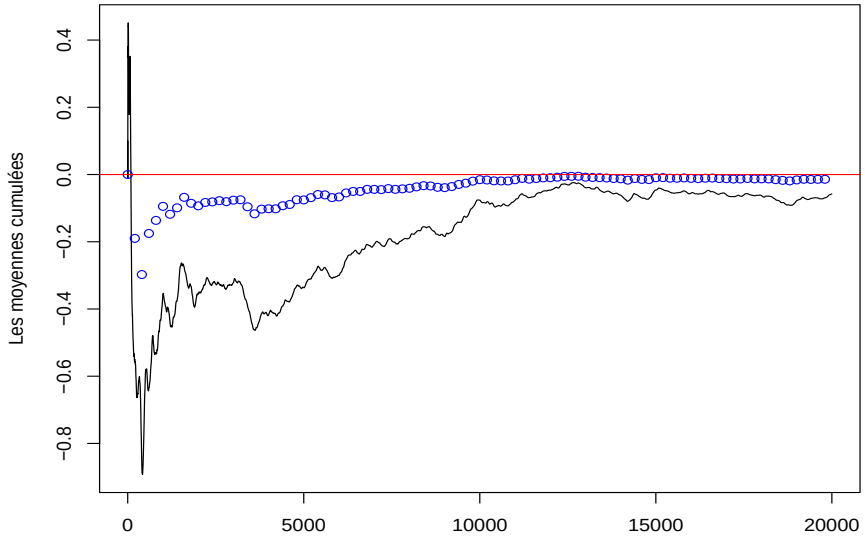

FIG 3.4. Comparaison entre l'estimateur modifié $\mu_{n,K}(F)$ et l'estimateur $\mu_n(F)$

Dans ce tableau nous présentons la comparaison entre les deux estimateurs, en estimant les facteurs (*l'estimateur* $\mu_n(F)$ *sur l'estimateur modifié* $\mu_{n,K}(F)$), simulés par $T = 200$ répétitions indépendantes et de même expérience, puis on calcule la variance des estimateurs $\mu_n(F)$ et $\mu_{n,K}(F)$ par:

$$\frac{1}{T-1}\sum_{i=1}^{T}\left[\mu_n^{(i)}(F)-\bar{\mu}_n(F)\right]^2 \quad \text{et} \quad \frac{1}{T-1}\sum_{i=1}^{T}\left[\mu_{n,K}^{(i)}(F)-\bar{\mu}_{n,K}(F)\right]^2$$

avec $\bar{\mu}_n(F)$ la moyenne des $\mu_n^{(i)}(F)$ et $\bar{\mu}_{n,K}(F)$ la moyenne des $\mu_{n,K}^{(i)}(F)$ respectivement.

Facteurs de réduction de variance						
	Nombre de simulation					
	$n = 1000$	$n = 10000$	$n = 50000$	$n = 100000$	$n = 200000$	$n = 500000$
Facteurs	3.54	25.12	129.83	255.8	423.92	1145.6

Tab 3.1. Facteurs de comparaison entre $\mu_{n,K}(F)$ et $\mu_n(F)$

Pour les résultats obtenus dans ce tableau, on remarque que le facteur de réduction de variance croît rapidement avec le nombre de simulation, et d'après ce que nous avons vu dans le deuxième chapitre, cette méthode possède un facteur de réduction de variance toujours très élevé.

Conclusion générale

Dans ce mémoire, nous avons vu de près quelques méthodes de réduction de variance permettant d'apporter des amélioration importantes pour beaucoup de résultats de calcul à travers des exemples simples que nous avons proposés avec les programmes correspondants ainsi que les résultats que nous avons obtenus, mais cette simplicité pose un problème incontournable car elle penche à dire que ces méthodes fonctionnent dans n'importe quelle situation, mais ça malheureusement n'est pas le cas, car chaque méthode a ses limites et ses difficultés d'application dans les modèles complexes.

En perspective, il serait intéressant de prendre ces méthodes dans un champ plus pratique, tout en montrant qu'est-ce qu'elles résous comme problème ? Qu'elles sont leurs limites et les difficultés qu'on peut rencontrer dans les modèles réalistes ? Dans quelle situation nous pouvons préférer une méthode par rapport aux autres ? Peut-on construire des algorithmes optimaux et efficaces ?

Bibliographie

[1] Boudiba Mohand Arezki. (2012). *Calcul de probabilité, Chaînes de Markov et Martingales, cours et exercices*. Cours de Master. Université de Mouloud Mammeri, Tizi-Ouzou.

[2] Benaïm Michel et El Karoui Nicole. (2007). *Promenade aléatoire*. Les éditions de l'école polytechnique.

[3] Boreux J.,Parent E. et Bernier J. (2010). *Pratique du calcul bayésien*. Springer.

[4] Brémaud Pierre. (2009). *Initiation aux Probabilités et aux chaînes de Markov*. Springer-Verlag, Berlin Heidelberg.

[5] Caumel Yves. (2011). *Probabilités et processus stochastiques*. Springer-Verlag, France.

[6] Dagpunar J.S. (2007). *Simulation and Monte Carlo.* John Wiley & Sons Ltd. England.

[7] Dellaportas Petros and Kontoyiannis Ioannis (2012). *Control variates for estimation based on reversible Markov chain Monte Carlo samplers*. Journal of the Royal Statistical Society. 74, Part 1, pp. 133-161.

[8] Delyon Bernard. (2012). *Simulation et modélisation.* Cours de deuxième année de master, Université Rennes I. Gelman, A. et Rubin, D.B. (1992) *Inference from iterative simulation using multiple sequences (avec discussion).* Statistical Science 7, 457-511.

[9] Geman, S. et Geman, D. (1984). *Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images.* IEEE Trans. Pattern Anal. Mach. Intell., 6, 721-741.

[10] Gilks W.R., Richardson S.T. and Spiegelhalter D.J. (1996). *Markov chain Monte Carlo in Practice.* Chapman and Hall, London.

[11] Glasserman P. (2004). *Monte Carlo Methods in Financial Engineering.* Springer, New York, USA.

[12] Graham Carl. (2004). *Chaînes de Markov*. Dunod.

[13] Haario H., Saksman E. et Tamminen J. (2001). *An adaptive Metropolis algorithm.* Bernoulli, 7, 223-242.

[14] Harris, T. (1956). *The existence of stationary measures for certain Markov processes.* In Proc. Srd Berkeley Symp. Math. Statis. Prob. 2, 113-124. University of California Press.

[15] Henderson S. (1997). *Variance Reduction Via an Approximating Markov Process.* Ph. D. thesis, Department of Operations Research, Stanford University, Stanford, CA. Hernandez-Lerma O. and Lasserre J.B. (2000). *Markov Chains and Invariant Probabilities.* Birkhauser Verlag. Basel, Boston, Berlin.

[16] Jacques J., Droesbeke, Fine J. et Saporta G. (2001). *Méthodes bayésiennes en statistique*. Editions TECHNIP, France.

[17] Korn R., Korn E. Kroisandt G. (2010). *Monte Carlo Methods and Models in Finance and Insurance.* Chapman & Hall /CRC. New York.

[18] Mengersen, K.L., Robert, C.P., Guihennec-Jouyaux, C. (1999). *MCMC Convergence Diagnostics : A Reviewww (with Discussion).* Pages 415-440 of :Bernardo, J.M., Berger, J.O., Dawid, A.P., Smith, A.F.M. (eds), Bayesian Stat. Oxford University Press.

[19] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H, Teller, E. (1953). *Equations of State Calculations by Fast Computing Machines.* J.Chem. Phys., 21, 1087-1091.

[20] Meyn, S.P. and Tweedie R.L. (2009). *Markov Chains and Stochastic Stability* (2nd ed.), London: Cambridge University Press. Published in the Cambridge Mathematical Library.

[21] Millet Annie. *Méthodes de Monte-Carlo.* Master 2, Laboratoire de Probabilités et Modèles Aléatoires, Universités Paris 6 et Paris 7.

[22] Mireuta Matei. (2011). *Etude de la performance d'un algorithme Metropolis-Hastings avec ajustement directionnel.* Mémoire en vue de l'obtention du grade de Maîtreès sciences en mathématiques. Université de Montréal.

[23] Parent Eric et Bernier Jacques. (2007). *Le raisonnement bayésien : Modélisation et inférence*. Springer.

[24] Philippe A. (1997). *Simulation output by Riemann sums.* J. Statist. Comput. Simul. 59: 295-314.

[25] Philippe A. and Robert C.P.(2000). *Riemann sums for MCMC estimation and convergence monitoring.* Statistics and Computing (2001) 11, 103-115.

[26] Plummer M., Best N., Cowles K. & Vines K. (2006). *CODA : convergence diagnosis and output analysis for MCMC.* R News, 6.

[27] Raftery, A.E. et Lewis, S. (1992a). *How many iterations in the Gibbs sampler?* In Bayesian Statistics 4, Berger J.O., Bernardo J.M. , A.P. Dawid and A.F.M. Smith (Eds.), 763-773. Oxford University Press, Oxford.

[28] Raftery, A.E. et Lewis, S. (1992b). *The Number of Iterations, Convergence Diagnostics and Generic Metropolis Algorithms.* Tech, report, Department of Statistics, U. of Washington, Seattle.

[29] Robert C.P. and Casella G. (1996). *Rao-Blackwellization of sampling schemes.* Biometrika 83: 81-94.

[30] Robert C.P. (1996). *Méthodes de Monte Carlo par chaînes de Markov.* Economica, Paris.

[31] Robert C.P. (1998). *Discretization and MCMC Convergence Assessment.* Springer-Verlag, New York. Lecture Notes in Statistics, Vol. 135.

[32] Robert C.P. et Casella George. (2004). *Monte Carlo Statistical Methods.* Springer-Verlag.

[33] Robert C.P. (2006). *Le Choix Bayésien.* Springer, Paris.

[34] Robert C.P. et Casella George. (2011). *Méthodes de Monte Carlo avec R.* Springer-Verlag, France.

[35] Roberts G.O. Rosenthal J. S. (2004). *General state space Markov chains and MCMC algorithms.* Probability Surveys, Vol. 1 (2004) 20-71, ISSN: 1549-5787.

[36] Rubinstein R.Y. and Kroese D.P. (2008). *Simulation and the Monte Carlo Method.* John Wiley & Sons. Canada.

[37] Gelman A. & Rubin D. (1992). *Inference from iterative simulation using multiple sequences (with discussion).* Statist. Sci., 7, 457-511.

[38] Ycart B. (2002). *Modèles et Algorithmes Markoviens, Mathématiques et Applications 39.* Springer Verlag.