

République Algérienne Démocratique et Populaire.  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.  
Université A. Mira- Béjaia  
Faculté des Sciences Exactes  
Département de Recherche Opérationnelle

*Mémoire de fin de cycle*  
*En vue d'obtention du diplôme Master*  
*Option : Modélisation Mathématique et Techniques de Décision*

*Thème*

# *Les méthodes de Monte Carlo : (MCMC et PMC). Applications*

**Préparé par :**

M<sup>elle</sup> Belaïd Nawel

M<sup>elle</sup> Djerroud Lamia

**Devant le Jury composé de :**

<b>Présidente :</b>	M <sup>me</sup>	LAGHA	Karima	M.C.B	Université de Béjaïa
<b>Rapporteurs :</b>	M <sup>r</sup>	ADJABI	Smail	Professeur	Université de Béjaïa
	M <sup>r</sup>	ZOUGAB	Nabil	M.C.B	Université de Tizi-Ouzou
<b>Examineur :</b>	M <sup>r</sup>	DJOUDER	Sofiane	Magister	Université de Béjaïa
<b>Examinatrices :</b>	M <sup>elle</sup>	ZIANE	Yasmina	Doctorante	Université de Béjaïa
	M <sup>elle</sup>	Amroun	Sonia	Magister	Université de Béjaïa

**Juin 2013**

## *Remerciements*

Nous tenant à remercier en premier lieu "DIEU" qui nous aide et nous donne la patience et le courage durant ces années d'étude.

Nos vifs remerciements à monsieurs Smail ADJABI et Nabil ZOUGAB, en tant que Directeurs de mémoire, qui sont toujours montrés à l'écoute et très disponibles tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'ils ont bien voulu nous consacrer et sans eux ce mémoire n'aurait jamais vu le jour.

On n'oublie pas nos parents pour leurs contribution, leurs soutien et leurs patience.

## *Dédicace*

Je dédie ce modeste travail à :

Ceux qui j'ai tant aimés avec beaucoup d'affection et je suis très fière de les avoir et tous les mots du monde ne peuvent exprimer l'amour et le respect que je leur porte.

Et ceux à qui je dois tant.

Mes très chers parents.

Mes frères : Nassim, Ghilas.

Ma sœur : Yassmina.

Ma grande mère et ma tante auxquelles je souhaite une longue vie.

Mon oncle, sa femme et ses enfants.

Mes cousins et cousines.

A mes très chers et précieux amis.

Enfin, A la mémoire de ma grande mère maternelle .

*LAMIA*

## *Dédicace*

Je dédie ce modeste travail à :

A ceux qui m'ont tout donné sans rien en retour.

A ceux qui m'ont encouragé et soutenu dans mes moments les plus  
difficiles.

Mes très chers parents.

Mes sœurs : Nabila et Naima.

Mes nièces : Aya, Anayis et Amilia.

Mes frères : Nabil et H .

Mes cousins et cousines

A mes amis.

*NAWEL*

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Introduction Générale</b>	<b>1</b>
<b>1 Notions de bases des méthodes MCMC et PMC</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Simulation de Monte Carlo . . . . .	3
1.2.1 Génération de nombres aléatoires . . . . .	4
1.2.2 Simulation des nombres suivants différentes lois de probabilités . . . . .	4
1.2.3 Méthode de Monte-Carlo classique . . . . .	6
1.2.4 Méthode de Monte-Carlo par échantillonnage pondéré . . . . .	7
1.3 Chaîne de Markov à espace d'état discret $E$ . . . . .	7
1.3.1 Noyau d'une chaîne de Markov . . . . .	8
1.4 Estimation Bayésienne . . . . .	10
1.4.1 Densité a priori . . . . .	10
1.4.2 Calcul de la loi a posteriori . . . . .	11
1.4.3 Estimateur de Bayes . . . . .	11
1.5 Conclusion . . . . .	12
<b>2 Méthode MCMC</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Algorithme de Metropolis-Hastings . . . . .	14
2.2.1 Metropolis-Hastings indépendant . . . . .	14
2.2.2 Metropolis-Hastings à marche aléatoire . . . . .	15

---

2.2.3	Metropolis-Hastings à une variable à la fois . . . . .	15
2.3	Echantillonneur de Gibbs . . . . .	16
2.4	Convergence de la méthode MCMC . . . . .	17
2.5	Difficultés des méthodes MCMC . . . . .	18
2.6	Conclusion . . . . .	18
<b>3</b>	<b>Méthode PMC</b>	<b>19</b>
3.1	Introduction . . . . .	19
3.2	Méthode d'échantillonnage préférentiel . . . . .	19
3.3	Algorithme PMC général . . . . .	20
3.4	Algorithme PMC à données manquantes . . . . .	21
3.5	Algorithme Rao-Blackwell à données manquantes . . . . .	22
3.6	Avantages de PMC . . . . .	23
3.7	Inconvénients et difficultés de PMC . . . . .	23
3.8	Conclusion . . . . .	24
<b>4</b>	<b>Comparaison entre MCMC et PMC</b>	<b>25</b>
4.1	Introduction . . . . .	25
4.2	Partie 1 : Simulation . . . . .	25
4.3	Modèle Gaussien . . . . .	25
4.3.1	Méthode MCMC . . . . .	25
4.3.2	Méthode PMC . . . . .	29
4.3.3	Discussion des résultats . . . . .	32
4.4	Modèle Binomial . . . . .	33
4.4.1	Méthode MCMC . . . . .	33
4.4.2	Méthode PMC . . . . .	35
4.4.3	Discussion des résultats . . . . .	37
4.5	Modèle log Normale . . . . .	37
4.5.1	Méthode MCMC . . . . .	37
4.5.2	Méthode PMC . . . . .	40
4.5.3	Discussion des résultats . . . . .	43
4.6	Modèle Poisson hiérarchique . . . . .	44
4.6.1	Méthode MCMC . . . . .	44

---

4.6.2	Méthode PMC . . . . .	48
4.6.3	Discussion des résultats . . . . .	51
4.7	Modèle de Bernoulli . . . . .	51
4.7.1	Méthode MCMC . . . . .	51
4.7.2	Méthode PMC . . . . .	54
4.7.3	Discussion des résultats . . . . .	55
4.8	Modèle Exponentiel à données manquantes . . . . .	56
4.8.1	Méthode MCMC . . . . .	56
4.8.2	Méthode PMC . . . . .	58
4.8.3	Discussion des résultats . . . . .	59
4.9	Partie 2 : Application réelle . . . . .	60
4.10	Modèle à changement de point . . . . .	60
4.11	Application et résultats . . . . .	61
4.12	Discussion . . . . .	73
	<b>Conclusion générale</b>	<b>74</b>
	<b>Annexe</b>	<b>76</b>
	<b>Bibliographie</b>	<b>96</b>

# Introduction Générale

La statistique s'est considérablement développée, initiant une révolution dans les modes de pensée, car elle porte un langage de représentation du monde et de ses incertitudes, mais les statisticiens ont découvert sur le tard, les avantages qu'ils pouvaient tirer de ses techniques. C'est aujourd'hui une science mathématique dont l'objectif est de décrire ce qui s'est produit et de faire des projections quand à ce qu'il peut advenir dans le futur. Elle possède une composante théorique ainsi qu'une composante appliquée. La composante théorique est proche de la théorie des probabilités et forme avec cette dernière, les sciences de l'aléatoire. La statistique appliquée est utilisée dans presque tous les domaines de l'activité humaine : ingénierie, management, économie, biologie, informatique, etc.

Les méthodes de Monte Carlo sont des méthodes statistiques d'intégration numérique, qui utilisent l'aléatoire. Elles permettent de résoudre de nombreux problèmes comme par exemple, l'évaluation d'intégrales sur des domaines complexes (et/ou) de grande dimension et l'optimisation. Les grandes avancées méthodologiques dans le domaine des méthodes de Monte Carlo sont souvent dûes aux physiciens.

Depuis une quinzaine d'années, on assiste à une forte progression de l'analyse bayésienne dans de nombreux domaines. En estimation bayésienne, souvent le calcul explicite de la loi a posteriori du vecteur des paramètres à estimer est impossible ainsi, les méthodes de Monte Carlo par chaînes de Markov(MCMC) ont été introduites par Metropolis en 1953 et al.[20] en physique statistique puis généralisées par Hastings en 1970 à la simulation statistique [15]. Il faut attendre l'article de A.E. Gelfand et A.F.M. Smith pour que l'utilisation de ces méthodes se développent en statistique appliquée et pour qu'elles connaissent, depuis lors, un essor extrêmement rapide, surtout en statistique et précisément en inférence par la simulation. Elles ont été très utilisées et ont permis des avancées notables dans le traitement bayésien de nouveaux modèles ; c'est notamment le cas de nombreux modèles à données manquantes.

L'algorithme de Metropolis-Hasting et l'échantillonneur de Gibbs sont les schémas les plus répandus de ces méthodes, dans le but de construire des chaînes de Markov ergodiques [14]. De par leurs ca-



ractère «universel» et leurs relative facilité d'implémentation, ces algorithmes sont couramment utilisés comme outils d'approximation et sont préférés aux méthodes d'échantillonnage préférentiel. Cependant, la convergence de la chaîne de Markov vers la loi stationnaire (cible) peut s'avérer très lente et surtout impossible à quantifier théoriquement. C'est ainsi qu'en tirant profit de l'expérience acquise lors du développement des méthodes MCMC tout en supprimant la difficulté fondamentale d'évaluation de la convergence, Cappé et al. [2004] ont proposé de généraliser la technique de l'échantillonnage préférentiel, qui a donné naissance à PMC (Population Monte Carlo), initialement pour traiter le problème de reconstruction de signal dans un canal ionique, puis amélioré par Douc et al. en (2005). Très récemment Celeux et al. en (2006) [5] ont également mis à profit cette méthode pour étudier des modèles à données manquantes .

Dans ce présent mémoire, nous comparons sur des données simulées et sur des données réelles les méthodes, Monte Carlo par Chaîne de Markov (MCMC) et Population Monte Carlo (PMC). Notre travail est organisé en cinq chapitres :

- Le chapitre 1 décrit quelques méthodes de simulation classiques, les méthodes de Monte Carlo qui sont des méthodes d'intégration numérique qui utilisent l'aléatoire, et la théorie des processus Markoviens dans le but d'étudier les algorithmes MCMC qui seront exposés dans le deuxième chapitre.
- Le chapitre 2 présente les principaux algorithmes MCMC : l'algorithme de Hastings-Metropolis et ses différents types, l'échantillonneur de Gibbs et enfin une méthode de diagnostic de convergence de ces algorithmes.
- Dans le chapitre 3 nous présentons l'algorithme d'échantillonnage préférentiel, et les algorithmes de Population Monte Carlo (PMC).
- Le chapitre 4 est consacré à la comparaison des méthodes MCMC et PMC. Nous appliquerons ces deux méthodes sur des modèles statistiques connus comme le modèle gaussien, binomial... etc, afin d'estimer les paramètres de ces modèles.
- Le dernier chapitre est consacré à une application réelle, dans lequel nous appliquerons les méthodes MCMC et PMC pour les comparer.
- Ce travail se termine par une conclusion générale, une bibliographie et une annexe dans laquelle sont présentés les différents programmes utilisés.

# Notions de bases des méthodes MCMC et PMC

## 1.1 Introduction

L'origine de la simulation repose sur le tirage des boules dans une urne, le tirage de cartes ou le jet d'un dé pour obtenir des tables de nombres aléatoires.

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

- L'universalité de son application et sa souplesse ;
- Elle permet d'obtenir les différentes caractéristiques du système, ce qui est particulièrement important quand l'opération se trouve au stade de conception.

## 1.2 Simulation de Monte Carlo

La méthode de Monte Carlo, initialement élaborée par Nicholas Metropolis et Stanislaw Ulam, a été utilisée en 1940 pour le projet Manhattan au laboratoires de Los Alamos. Elle est appliquée en 1966 à l'étude du transport électronique par Kurosawa.

Quand il s'agit des systèmes complexes, il s'avère que l'application des méthodes analytiques devient très compliquée, d'où la nécessité d'appliquer la méthode de Monte Carlo.

Une des phases les plus importantes de cette méthode est la génération de nombres aléatoires.

### 1.2.1 Génération de nombres aléatoires

La génération de nombres aléatoires est primordiale, car elle sera incluse dans le modèle et fournira au fur et à mesure, les échantillons artificiels d'entrée au simulateur.

L'objectif de la génération de nombres aléatoires est de produire une suite de nombres aléatoires indépendant et répartis uniformément sur l'intervalle  $[0, 1]$ .

### 1.2.2 Simulation des nombres suivants différentes lois de probabilités

La simulation de nombres aléatoires suivant une loi de probabilité est basée sur un générateur de nombre aléatoires uniformément distribués sur l'intervalle  $[0, 1]$ .

Il existe dans la littérature une multitude de méthodes pour générer des nombres suivant une loi de probabilité, le choix de l'une ou de l'autre dépend essentiellement de cette dernière.

#### Méthode d'inverse (Transformation)

Cette méthode est utilisée pour générer des nombres suivant des lois dont l'expression de la fonction de répartition est facilement inversible [9], telles que la loi exponentielle et la loi de Weibull.

Soit une variable aléatoire  $X$  dont la fonction de répartition est  $F(x) = \Pr(X \leq x)$ ; posons  $y = F(x)$ .

On peut démontrer que si  $Y$  est uniformément distribuée sur  $[0, 1]$  alors  $X = F^{-1}(Y)$  possède la fonction de répartition :

$F(x) = \Pr(Y \leq F(x))$ , donc pour une distribution donnée, il suffit de calculer  $y$  à partir de  $F$ , ce qui revient à déduire  $x$  à partir de  $y$ .

#### Exemple 1. (loi exponentielle)

Soit  $F(x)$  la fonction de répartition d'une variable aléatoire suivant une loi exponentielle de moyenne  $\frac{1}{\lambda}$ .

$$F(x) = 1 - \exp(-\lambda x)$$

On pose  $y = F(x)$

$$D'où  $x = \frac{-1}{\lambda} \log(y)$$$

Où  $y$  suit une loi uniforme sur l'intervalle  $[0, 1]$ .

## Méthode de rejet-acceptation

Si l'on désire générer des valeurs d'une variable aléatoire avec une densité  $f(x)$ , alors que ses propriétés ne sont pas connues, et sa fonction de répartition associée n'est pas inversible d'une manière explicite. La méthode de rejet-acceptation est utilisée s'il existe une fonction  $g(x)$  ayant un domaine identique à celui de  $f$  et une constante  $C$  positive [19], telle que :

$$f(x) \leq Cg(x) \quad , \quad \forall x \in \Theta.$$

La méthode se résume comme suit :

1. Générer un nombre  $y$  selon  $g$  ;
2. Générer un nombre  $u$  selon  $U_{[0,1]}$  ;
3. Si  $u$  vérifie la condition  $u \leq \frac{f(y)}{Cg(y)}$  alors :  
On accepte la valeur générée ;  
sinon on rejette  $u$  et  $y$ .

$C$  doit vérifier l'équation :  $C \geq f(x)/g(x)$ ,  $\forall x \in \Theta$ .

La valeur optimale de  $C$  est donc :

$$C = \max_{\Theta} f(x)/g(x).$$

Le taux d'acceptation sera alors défini par :

$$\rho = f(x)/Cg(x).$$

Remarquons que si  $g(x)$  est proche de zéro, alors  $C$  devient très grand et le taux d'acceptation moyen diminue. L'efficacité de l'algorithme dépend donc de l'adéquation entre  $f$  et  $g$ .

## Méthode pour la simulation d'un échantillon suivant la loi Normale

### Méthode d'approximation

Cette méthode repose sur le théorème central limite :

**Théorème 1.** Soient  $(z_1, z_2, \dots, z_k)$ ,  $k$  variables aléatoires indépendantes et identiquement distribuées telles que :  $E(z_i) = m$  et  $Var(z_i) = \sigma^2$ . Alors lorsque  $k \rightarrow +\infty$  la variable  $\frac{\sum_{i=1}^k z_i - km}{\sqrt{k}\sigma}$  suit une loi normale centrée réduite. Alors les variables  $y_j = \frac{\sum_{i=1}^k z_i - km}{\sqrt{k}\sigma}$  sont générées approximativement selon une loi normale centrée réduite.

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

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

sont générées selon une loi normale centrée réduite.

### Méthode de Box-Muller

La méthode de Box et Muller [9] produit des couples de nombres aléatoires exactes selon une loi normale centrée réduite.

Soient  $X$  et  $Y$  deux variables aléatoires indépendantes qui suivent une loi normale centrée réduite.

Soit  $f(x, y)$  la densité conjointe du couple  $(X, Y)$  :  $f(x, y) = \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}}$

On transforme  $x$  et  $y$  en coordonnée polaire :

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

#### Algorithme de Box-Muller :

- Généré  $u_1 \sim U_{[0,1]}$ ;
- Poser  $\theta = 2.\pi.u_1$ ;
- Généré  $u_2 \sim U_{[0,1]}$ ;
- Poser  $r = \sqrt{-2.\log(1 - u_2)}$ ;
- $x = r \sin \theta$ ;
- $y = r \cos \theta$ .

### 1.2.3 Méthode de Monte-Carlo classique

La méthode de Monte-Carlo classique consiste à estimer par simulation des espérances :

$$E_f[h(x)] = \int_{\Theta} h(x)f(x)dx. \tag{1.1}$$

Pour  $h$  donnée.

Si l'intégrale ne peut être calculée explicitement, alors on peut la calculer comme dans [18] :

Générer aléatoirement un échantillon de taille  $m$  ;  $x_1, \dots, x_m$  suivant la distribution  $f$ , puis calculer :

$$a_m = \frac{1}{m} \sum_{j=1}^m h(x_j).$$

Alors, par la loi des grands nombres,  $E_f[h(x)]$  est approximé par  $a_m$ .

L'échantillon généré suivant  $f$  peut être calculé par exemple à partir de la méthode rejet-acceptation ou la méthode d'inverse.

### 1.2.4 Méthode de Monte-Carlo par échantillonnage pondéré

L'échantillonnage pondéré suppose que pour approcher l'intégrale :

$$\int_{\Theta} h(x)f(x)dx.$$

On utilisera une autre fonction  $g$  appelée loi instrumentale pouvant être sélectionnée parmi les lois simple.

La méthode d'échantillonnage pondérée est toute méthode proposant d'approcher l'intégrale à partir de l'échantillon  $(x_1, \dots, x_m)$  généré suivant la loi  $g$ , par l'approximation donnée par [21] :

$$\frac{1}{m} \sum_{j=1}^m \frac{f(x_j)}{g(x_j)} h(x_j).$$

Cet estimateur, converge vers l'intégrale pour les mêmes raisons que la méthode de Monte-Carlo classique quel que soit le choix de la loi de  $g$  parmi les lois ayant un support contenant celui de  $f$ .

#### Choix de la distribution instrumentale

Le choix de  $g$  n'étant pas restreint, cependant, il existe une densité  $g$  optimale [3] qui minimise la variance théorique.

Certaines précautions sont cependant nécessaires :

- Si  $\sup \frac{f}{g} = \infty$ ,  $g$  doit être rejeté.
- Si  $\sup \frac{f}{g} = M$ , l'échantillonnage pondéré peut être utilisé mais quelques fois en concurrence avec la méthode d'acceptation-rejet directe sur  $f$ .

## 1.3 Chaîne de Markov à espace d'état discret $E$

Une chaîne de Markov est une suite  $X_n$  de variables aléatoires, telles que la loi conditionnelle de  $X_n$  sachant  $X_{n-1}, X_{n-2}, \dots$  est la même que celle de  $X_n$  sachant  $X_{n-1}$ , c'est à dire :

$$\Pr[X_n = j | X_{n-1} = i_{n-1}, X_{n-2} = i_{n-2}, \dots, X_0 = i_0] = \Pr[X_n = j | X_{n-1} = i_{n-1}]. \quad (1.2)$$

La chaîne de Markov est homogène si de plus on a :

$$\Pr[X_n = j | X_{n-1} = i] = \Pr[X_1 = j | X_0 = i] \quad (1.3)$$

**Matrice de transition [22]**

Soit :

$$P = \begin{pmatrix} p_{00} & p_{01} & p_{02} & \dots \\ p_{10} & p_{11} & p_{12} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$P$  est appelée matrice de passage (ou de transition) de la chaîne et les  $p_{ij}$  sont les probabilités de transitions.

Toute matrice de transition  $P = (p_{ij})_{i,j \in E}$  vérifie les propriétés suivantes :

- ▷ pour tout couple  $(i, j)$ , on a  $p_{ij} \geq 0$ ;
- ▷ pour tout  $i \in E$ , on a  $\sum_{j \in E} p_{ij} = 1$ .

La matrice  $P$  est appelée *matrice stochastique*.

**1.3.1 Noyau d'une chaîne de Markov**

La première notion utile est celle de noyau car une chaîne de Markov homogène est entièrement définie par :

- Son noyau ;
- La loi initiale de  $X_0$  (ou sa valeur initiale).

Le noyau est une fonction à deux variables qui caractérise les probabilités de transition de  $X_n$  à  $X_{n+1}$ .

La forme dite intégrale du noyau est :

$$K_n(X, A) = [X_{n+1} \in A | X_n = X]$$

$A$  étant un sous-ensemble mesurable du domaine  $X$  de  $X_{n+1}$ . Pour une chaîne homogène, on notera simplement  $K(X, A)$  au lieu de  $K_n(X, A)$ .

Ainsi, pour tout couple  $(X, A)$ ,  $K(X, \cdot)$  est une mesure de probabilité sur l'espace  $X$  indexée par la valeur  $X$  et  $K(\cdot, A)$  est une fonction mesurable comprise entre 0 et 1, indexée par les sous-ensembles mesurables de  $X$ . Dans la plupart des cas, la mesure de probabilité  $K(X, \cdot)$  aura une densité  $k(X, X')$  par rapport à une mesure support adéquate (indépendante de  $n$  pour une chaîne homogène) et telle que :

$$k(X, X') = [X_{n+1} = X' | X_n = X]$$

## Probabilités invariantes

Une probabilité  $\pi$  sur  $E$  est appelée probabilité invariante, ou probabilité stationnaire [1], d'une chaîne de Markov de matrice de transition  $P$  si  $\pi = P\pi$ .

## Réversibilité [10]

On dit qu'une chaîne de Markov de matrice de transition  $P$ , est réversible par rapport à la probabilité  $\pi$  si on a pour tous  $i, j \in E$ ,

$$\pi(i)P(i, j) = \pi(j)P(j, i) \quad (1.4)$$

Si une chaîne de Markov est réversible par rapport à la probabilité  $\pi$ , alors  $\pi$  est une probabilité invariante.

## Irréductibilité [10]

On dit qu'une chaîne de Markov, ou sa matrice de transition, est irréductible si pour tous  $i, j \in E$ , la probabilité partant de  $i$  d'atteindre  $j$  est strictement positive, autrement dit : si pour tous  $i, j \in E$ , il existe  $n = n_{i,j} \geq 1$  telle que  $P^n(i, j) > 0$ .

## Récurrence

Soit  $(X_n)_{n \geq 0}$  une chaîne de Markov homogène de matrice de transition  $P$ . On définit le temps de retour en  $i$  par [16] :

$$T_i = \inf\{n \geq 1; X_n = i\}.$$

- On dit que  $i$  est récurrent si  $\Pr_i(T_i < \infty) = 1$ .
- Un état  $i$  récurrent est dit récurrent positif si de plus  $\mathbb{E}_i[T_i] < \infty$
- Un état  $i$  récurrent est dit récurrent nul si  $\mathbb{E}_i[T_i] = \infty$ .
- Un état qui n'est pas récurrent est dit transitoire.

## Périodicité [10]

Soit  $X$  une chaîne de Markov de matrice de transition  $P$ . La période d'un état  $i \in E$  est le PGCD de  $n \in \mathbb{N}^*$ ;  $P^n(i, i) > 0$ . Un état est dit apériodique si sa période est 1, sinon il est dit périodique. Une chaîne de Markov est dite apériodique si tous ses états sont apériodiques.



## Ergodicité [2]

Une chaîne de Markov homogène, irréductible, récurrente positive et apériodique est dite ergodique.

## 1.4 Estimation Bayésienne

### 1.4.1 Densité a priori

Supposons que l'on dispose d'information supplémentaire sur le paramètre  $\theta$  en plus de l'information apportée par l'échantillon, cette information est modélisée à travers une loi de probabilité, appelée loi a priori. Sa densité est notée  $\pi(\theta)$ .

Le modèle statistique paramétrique bayésien consiste en la donnée d'une loi *a priori* et d'une loi d'observations. On appelle loi des observations, la loi conditionnelle de  $X$  sachant  $\theta$ . Sa densité est notée  $\pi(x|\theta)$  que la variable aléatoire  $X$  soit discrète ou continue. Si  $X$  est discrète,  $\pi(x|\theta)$  représente  $Pr(X = x|\theta)$ . On fera dans ce chapitre systématiquement l'hypothèse que, sachant  $\theta$ , les v.a.  $X_i$  sont indépendantes. Autrement dit on aura :

$$\pi(x|\theta) = \prod_{i=1}^n \pi(x_i|\theta).$$

### Modélisation de l'information a priori

L'une des caractéristiques fondamentales de l'analyse Bayésienne est l'utilisation des densités de probabilité décrivant l'état de connaissance ou d'ignorance concernant les paramètres avant la prise en compte des observations. Le choix d'une telle densité  $\pi(\theta)$  qui est dite *a priori* est le problème le plus délicat et le plus critique de l'analyse Bayésienne.

En effet, il est très rare que l'information *a priori* disponible soit suffisamment précise pour pouvoir déterminer exactement cette densité. il faut donc faire des approximations dont le choix aura des répercussions sur les inférences statistiques Bayésiennes.

On introduira deux types de lois *a priori* qui sont les plus courants :

Les densités *a priori* conjuguées et les densités *a priori* non informatives [26].

### Densités a priori conjuguées

**Définition 1.** *La loi des observations étant supposée connue, on se donne une famille  $\mathcal{F}$  de lois de probabilité sur  $\Theta$ . On suppose que la loi a priori appartient à  $\mathcal{F}$ . Si dans ces conditions, la loi a posteriori (qui sera introduite plus tard) appartient encore à  $\mathcal{F}$ , on dit que la loi a priori est conjuguée.*

**Exemple 2.** Soit  $x = (x_1, \dots, x_i, \dots, x_n)$  un vecteur d'observations. On suppose que :  $X_i|\theta$  suit une loi de Bernoulli de paramètre  $\theta$  et que la loi a priori est une loi Beta. Comme  $\theta|x$  suit aussi une loi Beta, on en déduit que la loi Beta est ici conjuguée.

### Densités a priori non informatives

**Définition 2.** Les densités a priori non informatives sont utilisées dans le cas ou on ne dispose d'aucune connaissance a priori avant la prise en compte des observations. Une des lois de proposition sur  $\theta$  en absence d'information a priori sur  $\theta$ , est la loi uniforme sur  $\theta$ . Ce choix repose sur l'équiprobabilité des valeurs possibles de paramètre sur son domaine de définition Ainsi la loi a priori de  $\theta$  :

$\pi(\theta) = k$ , où  $k$  est une constante.

### 1.4.2 Calcul de la loi a posteriori

On dispose d'un vecteur d'observations :  $x = (x_1, \dots, x_i, \dots, x_n)$ , et on considère  $\theta$  le paramètre à estimer. La densité a posteriori notée  $\pi(\theta|x)$  est définie à partir du théorème de Bayes :

$$\pi(\theta|x) = \frac{\pi(x|\theta) \pi(\theta)}{\int_{\Theta} \pi(x|\theta) \pi(\theta)d\theta};$$

où :  $\int_{\Theta} \pi(x|\theta) \pi(\theta)d\theta$  est une constante de normalisation appelée aussi évidence Bayésienne.

### 1.4.3 Estimateur de Bayes

**Définition 3.** En générale on appelle estimation bayésienne du paramètre  $\theta$  la moyenne de la loi a posteriori. Cette moyenne est notée  $E[\theta|x]$ . Formellement, [26] on a :

$$E[\theta|x] = \int_{\Theta} \theta \pi(\theta|x)d\theta = \frac{\int_{\Theta} \theta \pi(x|\theta) \pi(\theta)d\theta}{\int_{\Theta} \pi(x|\theta) \pi(\theta)d\theta}.$$

### Risque de Bayes [26]

Pour définir la propriété du risque, il faut introduire un critère d'écart entre  $\theta$  et la valeur de son estimateur  $\hat{\theta}$ , cet écart noté  $l(\theta, \hat{\theta})$  et appelé fonction coût ou encore fonction perte (c'est souvent l'écart quadratique  $(\theta - \hat{\theta})^2$ ). Il en déduit alors la fonction risque Bayésien  $R(\hat{\theta}, \theta)$  définie comme :

$$R(\hat{\theta}, \theta) = \int l(\hat{\theta}, \theta) \pi(\theta|x)d\theta.$$

On dira que  $\hat{\theta}_1$  est meilleur que  $\hat{\theta}_2$  au sens du risque de Bayes, si  $R(\hat{\theta}_1, \theta) < R(\hat{\theta}_2, \theta)$ .

**Définition 4.** Soient deux fonctions réelles  $f$  et  $g$  définies sur le même espace. On dit que  $f$  et  $g$  sont proportionnelles, ce qu'on note  $f \propto g$ , si il existe une constante  $a$  tel que  $f(y) = ag(y)$  pour tout  $y$ . La relation  $\propto$  est une relation d'équivalence.

**Remarque 1.** Dans le contexte Bayésien on a :  $\pi(\theta|x) \propto \pi(x|\theta)\pi(\theta)$ . En tant que fonctions de  $\theta$ , les deux expressions sont effectivement proportionnelles ; la constante  $a$  qui apparaît dans la définition est égale ici à  $\frac{1}{\int_{\Theta} \pi(x|\theta)\pi(\theta)d\theta}$  ; à noter que cette quantité est bien une constante, au sens où elle ne dépend pas de  $\theta$ .

## 1.5 Conclusion

Dans ce chapitre, nous avons présenté quelques notions concernant la simulation, les méthodes de Monte Carlo et les chaînes de Markov nécessaires à la compréhension des deux méthodes qu'on va présenter dans les chapitres qui suivent. Nous avons présenté également l'estimation Bayésienne.

# Chapitre 2

## Méthode MCMC

### 2.1 Introduction

La méthode MCMC est publiée en 1953 par Metropolis et ses coauteurs [20], elle est étendue en 1970 par Hastings[15]. En 1984, les frères Geman proposent l'échantillonneur de Gibbs pour la restauration bayésienne d'images ; Cet échantillonneur est développé par Gelfand et Smith[13].

C'est au début des années 90, après le développement et la démocratisation de l'outil informatique, que ces outils rencontrent un important succès [24].

En statistique bayésienne,  $\theta$  est considéré comme un vecteur aléatoire de densité  $\pi(\theta)$  (loi a priori sur  $\theta$ ). L'estimation bayésienne est basée sur le calcul de la loi a posteriori de  $\theta$  dont la densité est notée  $\pi(\theta|y)$ . Les estimateurs bayésiens sont de la forme  $\mathbb{E}_\pi(h(\theta))$  où  $\mathbb{E}_\pi$  est l'opérateur des espérances mathématiques pour la loi  $\pi(\theta|y)$ .

$$I_h = \mathbb{E}_\pi(h(\theta)) = \int h(\theta)\pi(\theta|y)d\theta. \quad (2.1)$$

Souvent, le calcul explicite de la densité  $\pi(\theta|y)$  n'est pas envisageable ; ainsi de nombreuses méthodes d'approximation ont été proposées et la plus utilisée est celle de MCMC dont le principe est de construire une chaîne de Markov ergodique de réalisations  $\theta^{(1)}, \dots, \theta^{(t)}, \dots$ , qui a pour distribution stationnaire  $\pi(\theta|y)$  ; ainsi l'estimateur qui sera obtenu est  $\hat{I}_h$  :

$$\hat{I}_h = \frac{1}{T} \sum_{t=1}^T h(\theta^{(t)}). \quad (2.2)$$

La convergence de  $\hat{I}_h$  vers  $I_h$  est assurée par l'ergodicité de la chaîne de Markov.

Parmi les méthodes MCMC, l'algorithme de Hastings-Metropolis (Hastings 1970) et l'algorithme de Gibbs (Geman et Geman 1984) sont les plus utilisées et ont donné lieu a de nombreux algorithmes dérivés.

## 2.2 Algorithme de Metropolis-Hastings

### Principe

Le principe est de construire une chaîne de Markov ergodique de loi stationnaire  $\pi(\theta|y)$  partant de  $\theta^0$ , on génère une chaîne  $\theta^t$  à partir d'un noyau de transition de loi stationnaire  $\pi(\theta|y)$ , qui garantit de plus la convergence en loi vers  $\pi(\theta|y)$ . Pour  $T$  "assez grand" on peut considérer  $\theta^T$  comme distribué suivant  $\pi(\theta|y)$  et obtenir ainsi un échantillon  $\theta^T$  qui est effectivement distribué suivant  $\pi(\theta|y)$ .

### Algorithme 1 [25]

L'algorithme de Metropolis-Hastings consiste à simuler un échantillon selon une distribution objectif  $\pi(\theta|y)$ , à partir d'une distribution conditionnelle  $q(y|\theta)$ . Pour la mise en œuvre de l'algorithme, on doit pouvoir simuler facilement à partir de la distribution  $q(\cdot|\theta)$ , et sa densité  $q(y|\theta)$  doit être disponible analytiquement (au moins à une constante multiplicative près) ou à défaut, le rapport  $\pi(\theta|y)/q(y|\theta)$  doit être fini.

- **Étape 1** Pour  $t=0$ , on initialise  $\theta^{(0)}$  le premier élément de la chaîne ;
- **Étape 2** Pour  $t \in \{1 \dots T\}$  ;
  - On génère  $u \sim U_{[0,1]}$  ;
  - On génère  $y_t$  selon  $q(\cdot|\theta^{(t-1)})$  ;
  - On calcul  $\rho(\theta^{(t)}, y_t) = \text{Min}\{1, \frac{\pi(y_t)}{\pi(\theta^{(t)})} \frac{q(\theta^{(t)}|y_t)}{q(y_t|\theta^{(t)})}\}$
  - Puis, on prend

$$\theta^{(t+1)} = \begin{cases} y_t, & \text{avec la probabilité } \rho(\theta^{(t)}, y_t) \text{ si } u < \rho(\theta^{(t)}, y_t) \\ \theta^{(t)}, & \text{avec la probabilité } 1 - \rho(\theta^{(t)}, y_t) \text{ sinon} \end{cases}$$

- **Étape 3**  $t=t+1$  et retourner à l'étape 2 ;
- **Étape 4** finalement calculer l'estimateur de Bayes  $\hat{I}_h = \frac{1}{T} \sum_{t=1}^T h(\theta^{(t)})$

### 2.2.1 Metropolis-Hastings indépendant

L'algorithme repose sur l'utilisation de la loi instrumentale  $q(y|\theta^{(t)})$  indépendamment de  $\theta^{(t)}$ , c'est une généralisation de l'algorithme d'acceptation-rejet.

**Algorithme 2 [25]**

1. Générer  $y_t$  selon  $q(y)$  ;

2. Prendre

$$\theta^{(t+1)} = \begin{cases} y_t, & \text{avec la probabilité } \text{Min}\left\{1, \frac{\pi(y_t)}{\pi(\theta^{(t)})} \frac{q(\theta^{(t)})}{q(y_t)}\right\} \\ \theta^{(t)}, & \text{sinon} \end{cases}$$

Malgré que, les variables  $y_t$  soient simulées indépendamment, l'échantillon résultant n'est pas indépendant et identiquement distribué, puisque la probabilité d'acceptation de  $y_t$  dépend de  $\theta^{(t)}$ .

**2.2.2 Metropolis-Hastings à marche aléatoire**

Cette variante prend en compte la valeur précédemment générée pour simuler la suivante, la loi instrumentale  $q(y|\theta)$  sera ainsi écrite  $q(y-\theta) = q(\theta-y)$ , c'est-à-dire que  $y_t$  peut s'écrire  $\theta^{(t)} + \epsilon_t$ , tel que  $\epsilon_t$  étant une perturbation aléatoire de  $q$ , indépendante de  $\theta^{(t)}$ , ainsi la chaîne de Markov générée associée à  $q$  est une marche aléatoire.

L'algorithme s'écrit :

**Algorithme 3 [25]**

1. Étant donné  $\theta^{(t)}$

2. Générer  $y_t \sim q(y - \theta^{(t)})$  .

3. On calcul  $\rho(\theta^{(t)}, y_t) = \text{Min}\left\{1, \frac{\pi(y_t)}{\pi(\theta^{(t)})} \frac{q(\theta^{(t)} - y_t)}{q(y_t - \theta^{(t)})}\right\}$

4. Poser

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

**2.2.3 Metropolis-Hastings à une variable à la fois**

Quand le paramètre à simuler est de dimension grande, on est contraint de trouver une densité instrumentale multidimensionnelle engendrant une chaîne ayant le comportement d'une chaîne de Markov. Pour cela on peut utiliser un algorithme dit à une variable à la fois.

Le principe est de simuler les composantes  $(\theta_1^{(t)}, \theta_2^{(t)}, \dots, \theta_d^{(t)})$  une par une. À chaque itération de l'algorithme, on fait évoluer  $d$  composantes  $\theta_i^{(t)}$  en utilisant  $d$  étapes de l'algorithme Hastings-Metropolis,

ce qui signifie que pour obtenir le nouveau vecteur  $\theta^{(t+1)}$  il faudra utiliser  $d$  densités instrumentales  $q_i(\cdot|\cdot)$ .

Pour simuler une composante  $\theta_i^{(t)}$  il faut utiliser la loi instrumentale  $q_i(y_i|\theta_{-i}^{(t+1)})$  et la loi cible  $\pi_i(\theta_i|\theta_{-i}^{(t+1)})$ , où :  $\theta_{-i}^{(t+1)} = (\theta_{-i}^{(t+1)}, \dots, \theta_{i-1}^{(t+1)}, \theta_{i+1}^{(t+1)}, \dots, \theta_d^{(t+1)})$  (tel que  $i = 1, \dots, d$ ).

On aura ainsi la probabilité d'acceptation de l'étape  $i$  :

$$\rho_i(\theta_i^{(t)}, \theta_{-i}^{(t+1)}, y_i) = \text{Min}\left\{1, \frac{\pi_i(y_i|\theta_{-i}^{(t+1)}) q_i(\theta_i^{(t)}|y_i, \theta_{-i}^{(t+1)})}{\pi_i(\theta_i^{(t+1)}|\theta_{-i}^{(t+1)}) q_i(y_i|\theta_i^{(t)}, \theta_{-i}^{(t+1)})}\right\}$$

#### Algorithme 4 [25]

1. MH  $(\pi_1(\theta_1|\theta_{-1}^{(t+1)}), q_1)$
2. MH  $(\pi_2(\theta_2|\theta_{-2}^{(t+1)}), q_2)$
3. ...
4. MH  $(\pi_d(\theta_d|\theta_{-d}^{(t+1)}), q_d)$

#### Propriétés :

Les algorithmes de MH ne génèrent pas d'échantillon indépendant et identiquement distribués, en particulier parce que la probabilité d'acceptation de  $y_t$  dépend de  $\theta^{(t)}$ .

#### Remarque :

Les algorithmes de Metropolis-Hastings peuvent théoriquement être utilisés pour simuler un vecteur aléatoire de dimension  $p$  en utilisant une densité instrumentale multidimensionnelle. Mais quand  $p$  est grand, ce choix est rarement fait en pratique car la convergence d'un tel algorithme serait extrêmement lente. En effet, plus la dimension de l'espace des paramètres est grande, et plus la proportion de candidats rejetés est importante. On préférera utiliser l'algorithme de Gibbs comme cadre général.

## 2.3 Echantillonneur de Gibbs

Plus généralement, si l'on peut écrire  $\theta_1, \dots, \theta_p$  une partition du vecteur des paramètres  $\theta$  pour le modèle étudié, et si l'on peut spécifier complètement les lois de comportement a posteriori conditionnelles alors on peut facilement simuler, pas par pas, des réalisations conditionnelles de  $\theta_1, \dots, \theta_p$ . En itérant le procédé un grand nombre de fois, la chaîne de Markov produite par ces simulations répétées a pour distribution stationnaire  $\pi(\theta|y)$ .

**Algorithme 5 [4]**

1. Initialiser  $\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_p^{(0)})$
2. A l'itération  $i$  : simuler
 
$$\theta_1^{(i+1)} \sim \pi_1(\theta_1 | \theta_2^{(i)}, \dots, \theta_p^{(i)})$$

$$\theta_2^{(i+1)} \sim \pi_2(\theta_2 | \theta_1^{(i+1)}, \dots, \theta_p^{(i)})$$

$$\theta_p^{(i+1)} \sim \pi_p(\theta_p | \theta_1^{(i+1)}, \dots, \theta_{p-1}^{(i+1)})$$
3.  $i \rightarrow i + 1$  et aller en 2.

Les densités conditionnelles  $\pi_i$  sont appelées conditionnelles complètes.

**Propriétés :**

- Taux d'acceptation égal à 1. Toutes les valeurs simulées sont acceptées.
- Nécessite de connaître les lois conditionnelles de  $\pi$ , d'où une connaissance préalable de certaines propriétés probabilistes ou analytiques de  $\pi$ .

## 2.4 Convergence de la méthode MCMC

### Phase de chauffe

Afin de réduire le biais de l'estimateur bayésien dû à l'effet des valeurs initiales, un nombre d'itérations  $T_0$  sont ignorées. Le nombre d'itérations  $T_0$  tel que  $T_0 < T$  est appelé phase de chauffe (burn in).

Plusieurs méthodes ont été proposées dans la littérature pour vérifier la convergence des méthodes MCMC.

Nous citons par exemple, le critère de inter-intra chaîne proposé par Gelman and Rubin [1992].

Ce critère consiste à construire  $M$  chaînes en parallèle avec différentes valeurs initiales, puis de calculer le facteur d'échelle  $\rho$  en fonction des variances inter-chaîne et intra-chaîne introduit par [12] :

Soit  $M$  : le nombre de chaînes de Markov construites ; et  $T$  est le nombre d'itérations pour estimer le paramètre  $\theta$ .

Les variances inter-chaîne  $B$  et intra-chaîne  $W$  pour ces  $M$  chaînes de Markov sont définies respectivement par :

$$B = \frac{T}{M-1} \sum_{m=1}^M (\bar{\theta}_m - \bar{\theta})^2,$$

et

$$W = \frac{1}{M} \sum_{m=1}^M \frac{1}{T-1} \sum_{t=1}^T (\theta_m^{(t)} - \bar{\theta}_m)^2,$$



Avec :

$$\left\{ \begin{array}{l} \bar{\theta}_m = \frac{1}{T} \sum_{t=1}^T \theta_m^{(t)}, \\ \bar{\theta} = \frac{1}{M} \sum_{m=1}^M \bar{\theta}_m, \end{array} \right.$$

où  $\theta_m^{(t)}$  est le t-ième échantillon de la m-ième chaîne. Finalement le facteur d'échelle est défini par :

$$\hat{\rho} = \frac{1}{W} \left( \frac{T-1}{T} W + \frac{1}{T} B \right).$$

Une valeur de  $\sqrt{\hat{\rho}}$  proche de 1 indique une bonne convergence de la chaîne.

## 2.5 Difficultés des méthodes MCMC

La méthode MCMC peut présenter des difficultés importantes [5] à savoir :

- La convergence : c'est-à-dire l'atteinte de l'équilibre de la chaîne de Markov peut être très lente, surtout lorsque  $\pi(\theta|y)$  est difficile à approximer par une densité de transition  $q$  facilement simulable.
- Par ailleurs, même si il ya la convergence, détecter le "temps de chauffe" est loin d'être évident.
- MCMC est couteuse en temps de calcul.

## 2.6 Conclusion

Nous avons présenté dans ce chapitre les différents algorithmes de l'approche MCMC ainsi une méthode de diagnostic de convergence de ces algorithmes.

# Chapitre 3

## Méthode PMC

### 3.1 Introduction

Population Monte Carlo est le très récent et puissant algorithme introduit initialement par Cappé et al. en (2004) pour traiter le problème de reconstruction de signal dans un canal ionique ; puis amélioré par Douc et al. en (2005) et très récemment Celeux et al. en (2006) ont également mis à profit cette méthode pour étudier des modèles à données manquantes [5].

PMC est une amélioration du schéma d'échantillonnage préférentiel classique en lui introduisant une dimension itérative assurant l'adaptation de la loi instrumentale  $q(y|\theta)$  à la densité cible  $\pi(\theta|y)$ . Cette adaptation est implémentée de façon séquentielle [5] : au pas  $t$  de l'algorithme,  $M$  réalisations de  $\theta$  sont simulées à partir d'une densité instrumentale courante dépendante des  $M$  réalisations du pas  $t - 1$ . Ainsi l'estimateur  $\hat{I}_h^T$  asymptotiquement non biaisé de (2.1) sera comme suit :

$$\hat{I}_h^T = \frac{1}{T} \sum_{t=1}^T \sum_{j=1}^M w_t^{(j)} h(\theta_t^{(j)}), \quad (3.1)$$

où  $w_t^{(j)}$  est le poids normalisé de la particule  $j$  à la date  $t$  obtenu à partir de l'algorithme qui sera introduit plus tard.

### 3.2 Méthode d'échantillonnage préférentiel

Les algorithmes d'échantillonnage préférentiel, ou d'importance, sont des méthodes de simulation de Monte-Carlo qui, à partir d'un échantillon de variables aléatoires indépendantes et identiquement distribuées  $\theta_1, \dots, \theta_M$  provenant d'une densité instrumentale ou d'importance  $q(\theta)$ , [5] proposent d'estimer

l'intégrale :

$$I_h = \int h(\theta) \frac{\pi(\theta|y)}{q(\theta)} q(\theta) d\theta; \quad (3.2)$$

par

$$\hat{I}_h = \sum_{i=1}^M w_i h(\theta_i); \quad (3.3)$$

où

$$w_i = \frac{r_i}{\sum_{j=1}^M r_j} \quad \text{et} \quad r_i = \frac{\pi(y|\theta_i)\pi(\theta_i)}{q(\theta_i)}. \quad (3.4)$$

Où  $\pi(y|\theta_i)$  est la vraisemblance.

Ce type de méthodes peut apparaître fragiles dans des cas pratiques, l'estimateur  $\hat{I}_h$  étant fortement dépendant des poids normalisés  $w_i$ .

Tout l'art de l'échantillonnage préférentiel consiste donc à choisir une densité  $q$  qui doit être proche de la loi cible.

Différentes méthodes de construction de  $q$  ont été proposées. On peut les séparer en deux catégories à savoir :

1. Les méthodes statiques, où  $q(\theta)$  est déterminée de façon unique en préalable à la simulation.
2. Les méthodes adaptatives, c'est la méthode PMC.

### 3.3 Algorithme PMC général

Fondamentalement, [14] l'algorithme PMC général consiste en l'itération de deux étapes :

1. D'abord, on simule une population de  $n$  variables aléatoires (parfois dites particules) suivant une loi choisie a priori.
2. On calcule le poids de chaque particule suivant (3.4) et on retire un échantillon dans la population générée proportionnellement aux poids calculés.

Soit  $q_{it}$  ( $i = 1, \dots, M$  et  $t = 1, \dots, T$ ) une distribution d'importance. La représentation algorithmique qu'on va présenter correspond à ces deux étapes.

**Algorithme 6**

1. **Étape 0** : Choix de  $(\theta_0^{(1)}, \dots, \theta_0^{(M)})$ ;
2. **Étape t** ( $t=1, \dots, T$ ) :  
 Pour  $i=1, \dots, M$  :
  - Générer  $\theta_t^{(i)} \sim q_{it}(\theta)$  et calculer  $r_t^{(i)} = \frac{\pi(y|\theta_t^{(i)})\pi(\theta_t^{(i)})}{q_{it}(\theta_t^{(i)})}$ ;
  - Calculer  $\omega_t^{(i)} = r_t^{(i)} / \sum_{k=1}^M r_t^{(k)}$  et rééchantillonner les  $\theta_t^{(i)}$  en utilisant les poids  $\omega_t^{(i)}$ ;
  - Construire  $q_{i(t+1)}$  à partir de l'échantillon courant ;
3. Après avoir itéré  $T$  fois l'algorithme, un estimateur asymptotiquement non biaisé de l'intégrale (2.1) est donné dans la formule (3.1).

**3.4 Algorithme PMC à données manquantes**

La structure des données observables  $y$  induit [5] en général l'existence d'un ensemble de données manquantes. La distribution des données  $y$  peut être représentée par la densité suivante [14] :

$$\pi(y|\theta) = \int_{\mathbb{Z}} g(y, z|\theta) dz; \quad (3.5)$$

où  $z \in \mathbb{Z}$  représente le vecteur des données non observées.

De tels modèles ont été à l'origine de nombreux développements méthodologiques en statistique mathématique (algorithme «Expectation Maximisation» [Dempster et al., 1977, McLachlan and Krishnan, 1997] et l'algorithme de «Data Augmentation» [Tanner and Wong, 1987, Gelfand and Smith, 1990].)

On désigne par  $k(z|y, \theta)$  et  $\pi(\theta|y, z)$  les lois de distribution de  $z|y, \theta$  et  $\theta|y, z$  respectivement.

On utilise comme fonction d'importance :

$$q_{it}(\theta, z) = \pi(\theta|y, z)k(z|y, \theta).$$

Cette distribution consistant à simuler  $z_t^{(i)}$  suivant  $k(z|y, \theta_{t-1}^{(i)})$  et  $\theta_t^{(i)}$ , suivant  $\pi(\theta|y, z_t^{(i)})$ .

Finalement l'algorithme prend la forme suivante :

**Algorithme 7**

1. **Étape 0** Choix de  $(\theta_0^{(1)}, \dots, \theta_0^{(M)})$ ;
2. **Étape t** ( $t=1, \dots, T$ ) :  
 Pour  $i=1, \dots, M$  :

- Générer  $z_t^{(i)} \sim k(z|y, \theta_{t-1}^{(i)})$ ;
  - Générer  $\theta_t^{(i)} \sim \pi(\theta|y, z_t^{(i)})$ ;
  - Calculer  $r_t^{(i)} = \frac{g(y, z_t^{(i)}|\theta_t^{(i)})\pi(\theta_t^{(i)})}{k(z_t^{(i)}|y, \theta_{t-1}^{(i)})\pi(\theta_t^{(i)}|y, z_t^{(i)})}$ ;
  - Calculer  $\omega_t^{(i)} = r_t^{(i)} / \sum_{k=1}^M r_t^{(k)}$  et rééchantillonner les  $\theta_t^{(i)}$  en utilisant les poids  $\omega_t^{(i)}$ ;
3. Après avoir itéré  $T$  fois l'algorithme, un estimateur asymptotiquement non biaisé de l'intégrale (2.1) est donné dans la formule (3.1).

### 3.5 Algorithme Rao–Blackwell à données manquantes

Dans le but de diminuer la grande variabilité des poids due aux données manquantes donnée par l'algorithme 7.

Pour séparer la simulation des données manquantes et des valeurs courantes du paramètre  $\theta$  Rao–Blackwell [8] a considéré une approximation de la densité instrumentale de  $\theta_t^{(i)}$  par :

$$q_{it}(\theta) = \int q_{it}(\theta, z) dz = \int \pi(\theta|y, z) k(z|y, \theta) dz \quad (3.6)$$

Le principe de l'algorithme on réutilisant l'ensemble des données manquantes qui ont été simulées, est le suivant :

Soit :

$$q_{it}(\theta_t^{(i)}) \simeq d_t^{(i)} = \frac{1}{M} \sum_{l=1}^M \frac{k(z_t^{(l)}|y, \theta_{t-1}^{(i)})\pi(\theta_t^{(i)}|y, z_t^{(l)})}{k(z_t^{(l)}|y, \theta_{t-1}^{(l)})}.$$

Les poids  $r$  sont donnés en approximant à la fois les distributions a posteriori et instrumentales :

$$r_t^{(i)} = \frac{n_t^{(i)}}{d_t^{(i)}},$$

où

$$n_t^{(i)} = \frac{1}{M} \sum_{l=1}^M \frac{g(y, z_t^{(l)}|\theta_t^{(i)})\pi(\theta_t^{(i)})}{k(z_t^{(l)}|y, \theta_{t-1}^{(l)})}.$$

L'algorithme se résume en étapes suivantes :

**Algorithme 8**

1. **Étape 0** Choix de  $(\theta_0^{(1)}, \dots, \theta_0^{(M)})$ ;
2. **Étape t** ( $t=1, \dots, T$ ) :
  - (a) Pour  $i=1, \dots, M$  :
    - Générer  $z_t^{(i)} \sim k(z|y, \theta_{t-1}^{(i)})$ ;
    - Générer  $\theta_t^{(i)} \sim \pi(\theta|y, z_t^{(i)})$ ;
  - (b) Pour  $i=1, \dots, M$  :
    - Calculer  $n_t^{(i)} = \frac{1}{M} \sum_{l=1}^M \frac{g(y, z_t^{(l)} | \theta_t^{(i)}) \pi(\theta_t^{(i)})}{k(z_t^{(l)} | y, \theta_{t-1}^{(l)})}$ ;
    - Calculer  $d_t^{(i)} = \frac{1}{M} \sum_{l=1}^M \frac{k(z_t^{(l)} | y, \theta_{t-1}^{(i)}) \pi(\theta_t^{(i)} | y, z_t^{(l)})}{k(z_t^{(l)} | y, \theta_{t-1}^{(l)})}$ ;
    - Calculer  $r_t^{(i)} = \frac{n_t^{(i)}}{d_t^{(i)}}$  et  $\omega_t^{(i)} = r_t^{(i)} / \sum_{k=1}^M r_t^{(k)}$ ;
  - (c) Rééchantillonner les  $\theta_t^{(i)}$  en utilisant les poids  $\omega_t^{(i)}$ ;
3. Après avoir itéré  $T$  fois l'algorithme, un estimateur asymptotiquement non biaisé de l'intégrale (2.1) est donné dans la formule (3.1).

**Remarque [8]** Si :

- $M$  est très grand ( $M \geq 100.000$ ),  $T$  peut être pris égal à 2 ou 3.
- $M$  modérées ( $1000 \leq M \leq 10.000$ ),  $T$  doit être augmenté à  $T=10$ .

### 3.6 Avantages de PMC

La méthode PMC peut présenter des avantages [6] et [8] à savoir :

- Elle permet une grande généralité de choix instrumentaux à chaque pas de l'algorithme;
- Permet d'aboutir potentiellement à une meilleure efficacité que les algorithmes classiques d'échantillonnage préférentiel;
- Convergence rapide par rapport aux autres méthodes;
- L'estimateur obtenu est asymptotiquement non biaisé.

### 3.7 Inconvénients et difficultés de PMC

La méthode PMC peut présenter des inconvénients [6], [8] et [11] à savoir :

- L'inconvénient le plus habituel est la dégénérescence des particules;

- L'algorithme de Rao–Blackwell engendre un coût qui a tendance à ralentir l'algorithme ;
- Le choix du temps d'arrêt  $T$  de l'algorithme est un problème important qui est en cours d'étude.

## 3.8 Conclusion

Ce chapitre est consacré à l'illustration des algorithmes de PMC, ses avantages et inconvénients. Ces deux méthodes qu'on a présenté dans ce chapitre et le deuxième vont être appliquées afin de les comparer. Cette comparaison fera l'objet du chapitre suivant.

# Chapitre 4

## Comparaison entre MCMC et PMC

### 4.1 Introduction

Dans ce chapitre, nous allons présenter quelques modèles, afin de comparer les méthodes MCMC et PMC sur des données simulées dans la première partie. Dans la deuxième partie on traite un exemple d'application réel des méthodes MCMC et PMC, où nous définissons d'abord une notion nécessaire à la compréhension de cette application ; ensuite, nous présentons l'exemple traité avec MCMC et PMC et nous exposons les résultats de chaque méthode et leurs comparaisons.

### 4.2 Partie 1 : Simulation

### 4.3 Modèle Gaussien

#### 4.3.1 Méthode MCMC

On veut estimer la moyenne et l'écart type d'une loi *Normale* par l'approche bayésienne à partir d'un échantillon  $\{X_1, X_2, \dots, X_n\}$  distribué selon cette loi.  $X_i \sim \mathcal{N}(m, \sigma^2)$ .

On suppose que la loi a priori de  $m$  est gaussienne de moyenne  $\mu_0$  et de variance  $\sigma_0^2$  et la loi a priori de  $\sigma^2$  est une inverse gamma de paramètre  $\alpha$  et  $\beta$ , c'est à dire  $m \sim \mathcal{N}(m_0, \sigma_0^2)$  et  $\sigma^2 \sim \mathcal{IG}(\alpha, \beta)$ .

On peut montrer on utilisant la formule de Bayes que la loi a posteriori de  $m$  est une loi normale et celle de  $\sigma^2$  une inverse gamma et leurs paramètres sont donnés respectivement par :



- La moyenne

$$m|\sigma^2, x \sim \mathcal{N}(M, \Sigma^2);$$

avec

$$M = \frac{\frac{\sigma^2}{n}}{\frac{\sigma^2}{n} + \sigma_0^2} \mu_0 + \frac{\sigma_0^2}{\frac{\sigma^2}{n} + \sigma_0^2} \bar{x};$$

et

$$\Sigma^2 = \frac{1}{\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}};$$

où  $\bar{x} = \sum_{i=1}^n x_i$ .

- La variance

$$\sigma^2|m, x \sim \mathcal{IG}\left(\frac{n}{2} + \alpha, \frac{1}{2} \sum_{i=1}^n (x_i - m)^2 + \beta\right);$$

### Algorithme

L'algorithme utilisé est celui d'échantillonneur de Gibbs, il est appliqué sur un échantillon  $X$  généré selon la loi normale centrée réduite :

- Générer  $X \sim \mathcal{N}(0, 1)$ ;
- Initialiser les paramètres  $\mu_0, \sigma_0^2, \alpha, \beta$ ;
- À une itération  $t$  :
- Générer  $m^{(t)} \sim \mathcal{N}(M, \Sigma^2)$ ;
- Générer  $\sigma^{2(t+1)} \sim \mathcal{IG}\left(\frac{n}{2} + \alpha, \frac{1}{2} \sum_{i=1}^n (x_i - m^t)^2 + \beta\right)$ .

### Résultats

Les estimateurs de la moyenne et de la variance obtenus en générant 500 variables aléatoires  $X \sim \mathcal{N}(0, 1)$ , avec un nombre de période de chauffe de 3000 sur 10000 itérations sont :

$m$	$\sigma^2$
0.0363304	1.009834

Les représentations graphiques de ces résultats sont données dans les figures (4.1), (4.2), (4.3) et (4.4) :

## La moyenne

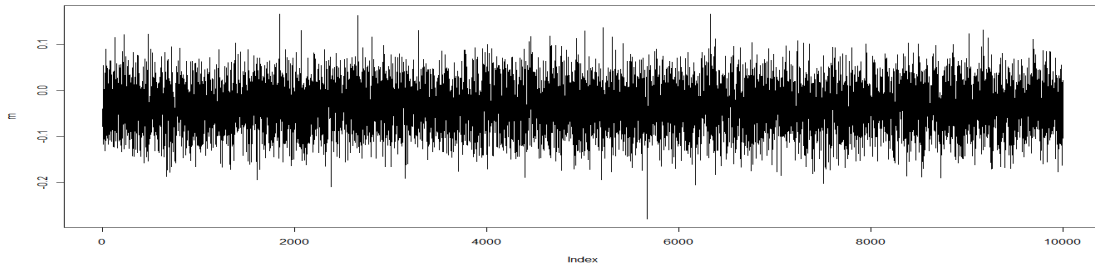


FIGURE 4.1 – Evolution de la chaîne de la moyenne ( $m$ )

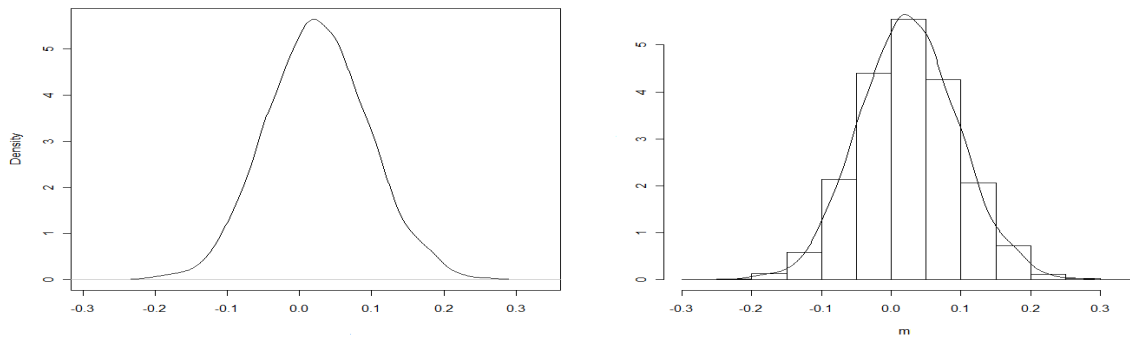


FIGURE 4.2 – Densité et histogramme de la chaîne de la moyenne ( $m$ )

## La variance

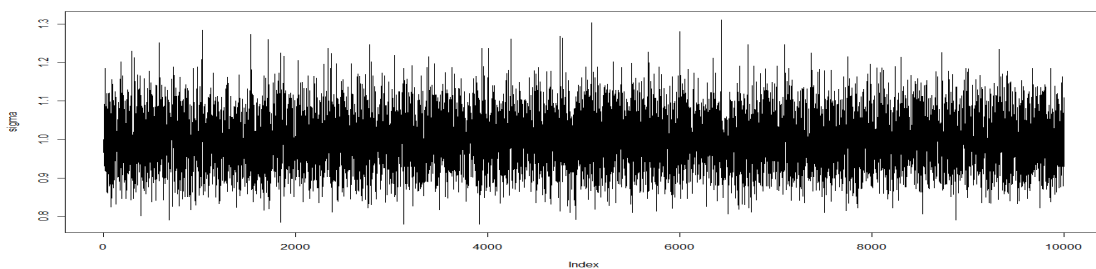
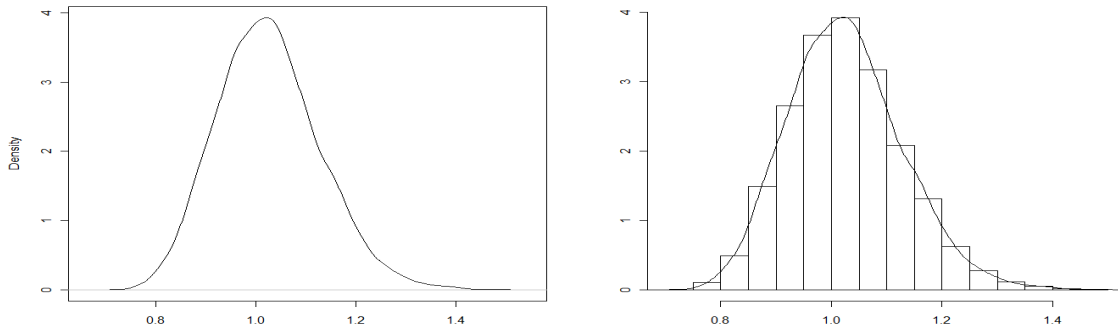


FIGURE 4.3 – Evolution de la chaîne de la variance ( $\sigma^2$ )


 FIGURE 4.4 – Densité et histogramme de la chaîne de la variance ( $\sigma^2$ )

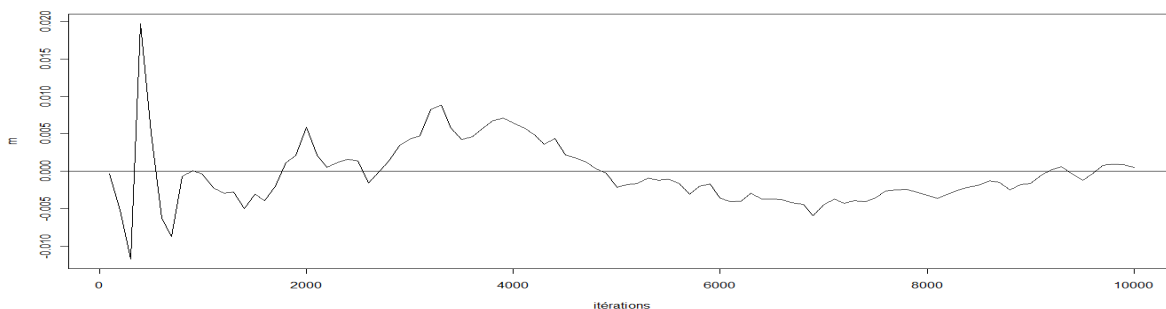
### Etude de la convergence

Pour vérifier la convergence de la méthode MCMC, nous avons utilisé le critère inter/intra chaîne de Gelman and Rubin [1992] dont le principe est donné dans le deuxième chapitre. Nous avons calculé le facteur d'échelle en utilisant  $M = 5$  et pour le nombre d'itération  $N = (1000, 5000, 10000)$ . Les résultats pour la moyenne sont donnés comme suit :

Nombre de chaînes	Nombre d'itérations	Facteur d'échelle
5	1000	1.0184
5	5000	1.006318
5	10000	1.000648

 TABLE 4.1 – Facteur d'échelle pour la moyenne ( $m$ )

La représentation graphique correspondante à la convergence de l'estimateur de la moyenne  $m$  est dans la figure suivante :

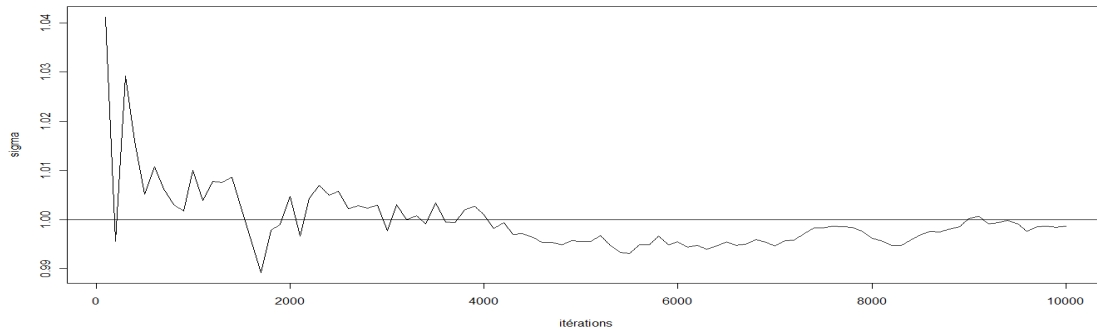

 FIGURE 4.5 – Convergence de l'estimateur de la moyenne ( $m$ )

Les résultats pour la variance sont donnés dans le tableau suivant :

Nombre de chaînes	Nombre d'itérations	Facteur d'échelle
5	1000	1.00984
5	5000	1.001845
5	10000	1.000397

 TABLE 4.2 – Facteur d'échelle de la variance ( $\sigma^2$ )

La représentation graphique correspondante à la convergence de l'estimateur de la variance  $\sigma^2$  est dans la figure suivante :


 FIGURE 4.6 – Convergence de l'estimateur de ( $\sigma^2$ )

### 4.3.2 Méthode PMC

#### Algorithme

L'algorithme utilisé est celui de PMC général qui consiste à :

- Générer  $X \sim \mathcal{N}(0, 1)$ ;
- Initialiser les paramètres  $\mu_0, \sigma_0^2, \alpha, \beta$ ;
- Générer un vecteur initial  $m \sim \mathcal{N}(0.2, 0.12)$  de taille  $M$  donnée;
- Générer un vecteur initial  $\sigma^2 \sim \mathcal{N}(0.2, 0.12)$  tronquée sur  $[0, +\infty[$  de taille  $M$ ;
- Les lois de propositions sont la loi *Normale* pour  $m$  et *Normale* tronquée sur  $[0, \infty[$  pour  $\sigma^2$ ;
- À l'étape  $t$  : générer un vecteur de taille  $M$  pour la moyenne ainsi que pour la variance selon les lois de propositions, calculer le poids de chaque particule ainsi les poids normalisés et rééchantillonner selon ces derniers .

## Résultats

Les estimateurs de la moyenne  $m$  et de la variance  $\sigma^2$  obtenus en générant 500 variables aléatoires  $X \sim \mathcal{N}(0, 1)$ , et  $T = 20$  sont :

$m$	$\sigma^2$
-0.002633846	1.023469

Les représentations graphiques des résultats sont données dans les figures (4.7), (4.8), (4.10) et (4.11) :

### La moyenne

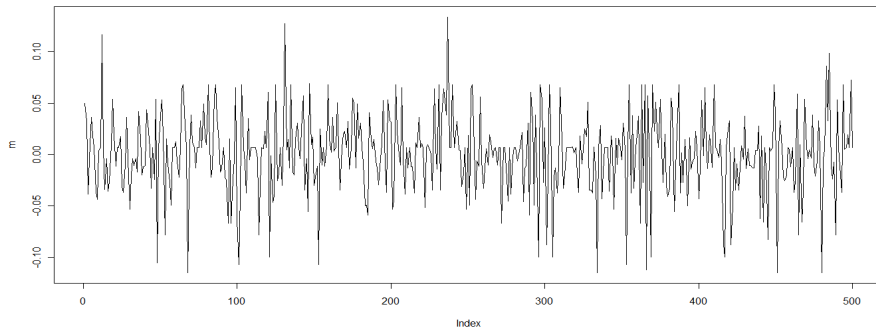


FIGURE 4.7 – Evolution de la chaîne de la moyenne ( $m$ )

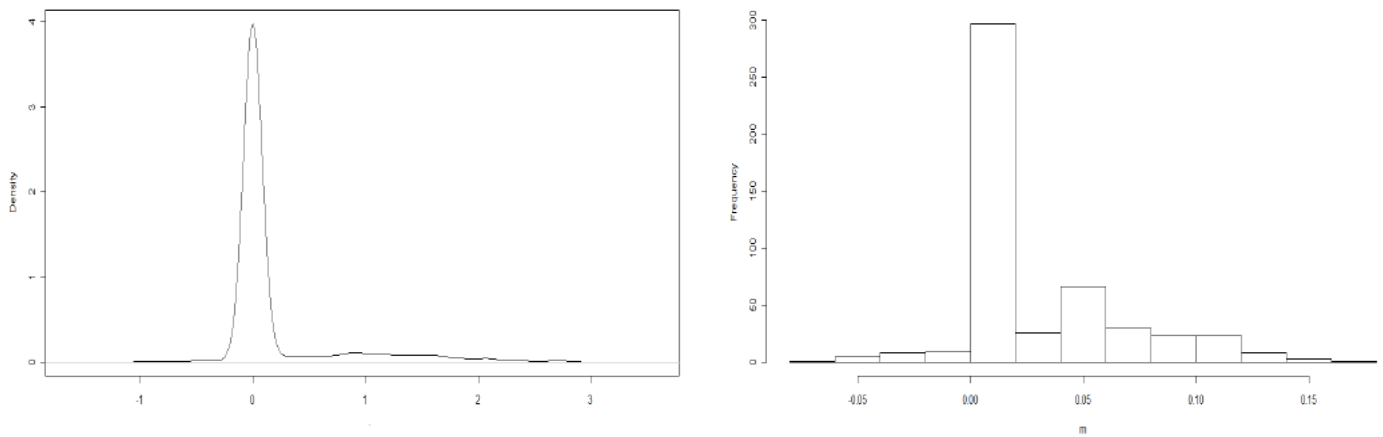


FIGURE 4.8 – Densité et histogramme de la chaîne de la moyenne ( $m$ )

La convergence de l'estimateur de la moyenne est donnée dans la figure suivante :

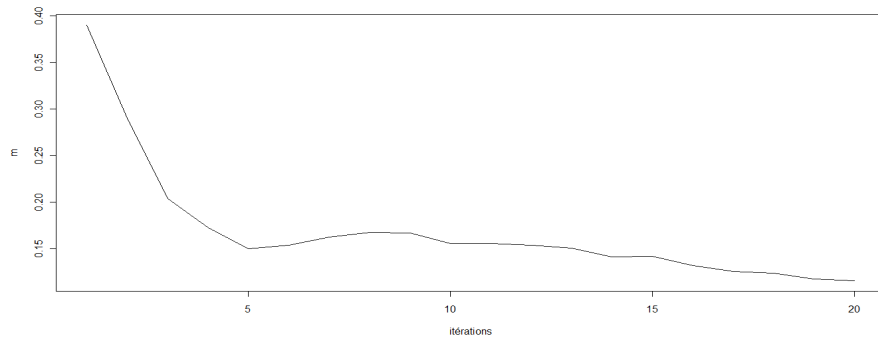


FIGURE 4.9 – Convergence de l'estimateur de la moyenne ( $m$ )

### La variance

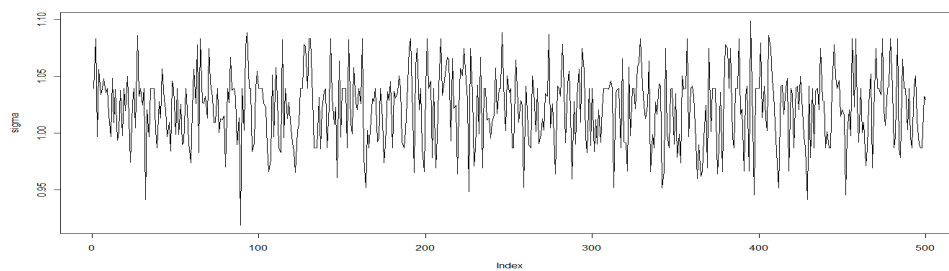


FIGURE 4.10 – Evolution de la chaîne de la variance ( $\sigma^2$ )

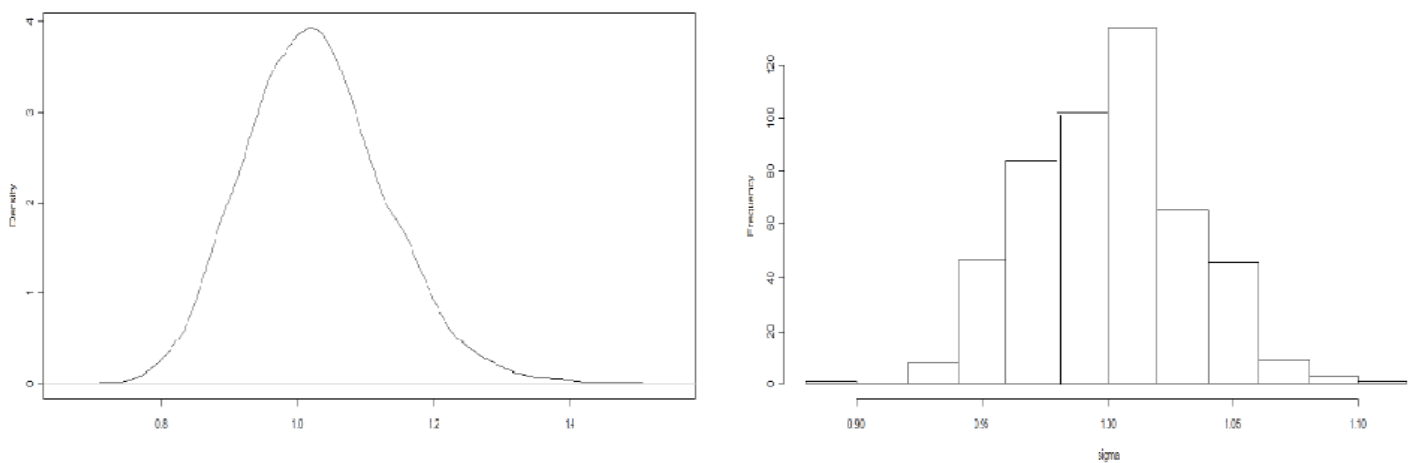


FIGURE 4.11 – Densité et histogramme de la chaîne de la variance ( $\sigma^2$ )

La convergence de l'estimateur de la variance est donnée dans la figure suivante :

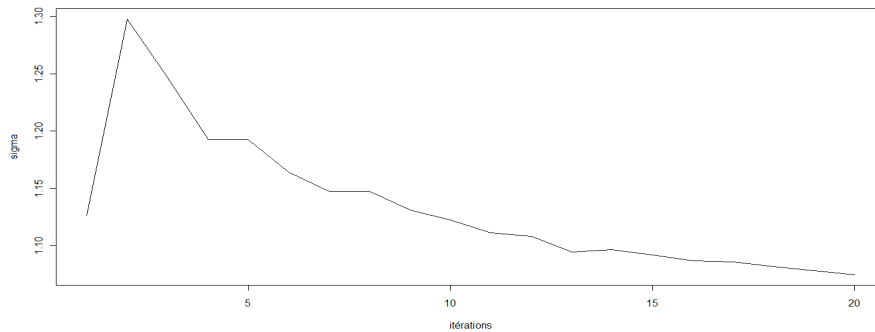


FIGURE 4.12 – Convergence de l'estimateur de la variance ( $\sigma^2$ )

### 4.3.3 Discussion des résultats

Les résultats montrent que pour la méthode MCMC, la chaîne de Markov obtenue pour la moyenne converge vers la valeur exacte du paramètre du modèle à partir d'un nombre d'itérations ( $> 5000$ ) et pour la variance à partir de ( $> 2000$ ) contrairement à la méthode PMC, pour un nombre de simulations relativement réduit ( $T = 15$ ) cette dernière fournit d'excellentes approximations des deux paramètres. Pour le temps d'exécution en fonction de la taille de l'échantillon nous avons obtenu les résultats suivants :

	MCMC (Gibbs)			PMC		
La taille de l'échantillon	$m$	$\sigma^2$	temps(s)	$m$	$\sigma^2$	temps(s)
50	-0.107	0.827	1.43	0.1547888	1.023983	0.64
100	0.04505	0.8738	1.45	0.04298319	1.062552	0.83
200	0.0371	0.949	1.50	-0.01827408	1.087073	1.18
500	0.01377	1.00498	1.68	-0.002633846	1.023469	1.5

TABLE 4.3 – Temps d'exécution de MCMC et PMC en secondes

Ces résultats montrent que la méthode PMC fournit d'excellentes approximations pour une durée inférieure à celle de la méthode MCMC.

## 4.4 Modèle Binomial

### 4.4.1 Méthode MCMC

Nous allons utiliser la méthode MCMC pour estimer le paramètre de la loi binomiale de densité :

$$f(x|n, p) = C_n^x p^x (1 - p)^{n-x}.$$

#### Algorithme [23]

L'algorithme utilisé est celui de Metropolis-Hastings à marche aléatoire. Pour  $n = 100$ ,  $x = 39$  et  $p = 0.4$  on a :

$$\pi(p|x) \propto p^{39}(1 - p)^{61} \tag{4.1}$$

- La loi de proposition est une *loi uniforme* ;
- À l'étape  $t$  générer  $p' \sim \mathcal{U}[0, (p^{(t-1)} + 1)]$
- Calculer la probabilité d'acceptation  $\rho(p^{(t)}, (p')) = \text{Min}[1, \frac{(p')^{39}(1-p')^{61} * \text{dunif}(p^{(t)}, 0, (p'+1))}{(p^{(t)})^{39}(1-p^{(t)})^{61} * \text{dunif}(p', 0, (p^{(t)}+1))}]$
- Générer  $u \sim \mathcal{U}[0, 1]$  et prendre :

$$p^{(t+1)} = \begin{cases} (p') & \text{avec la probabilité } \rho(p^{(t)}, (p')) \quad \text{si } u < \rho(p^{(t)}, (p')) \\ p^{(t)} & \text{avec la probabilité } 1 - \rho(p^{(t)}, (p')) \quad \text{sinon} \end{cases}$$

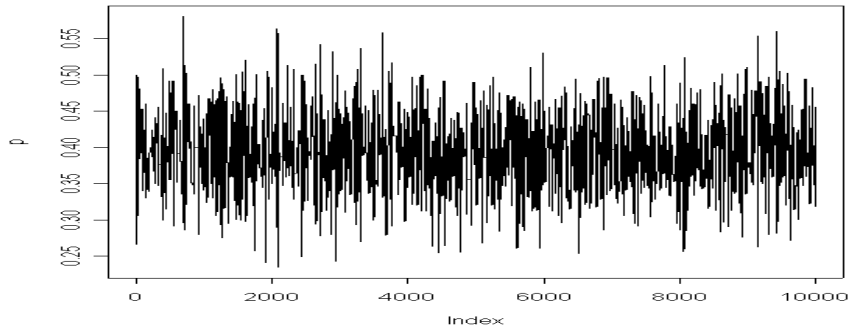
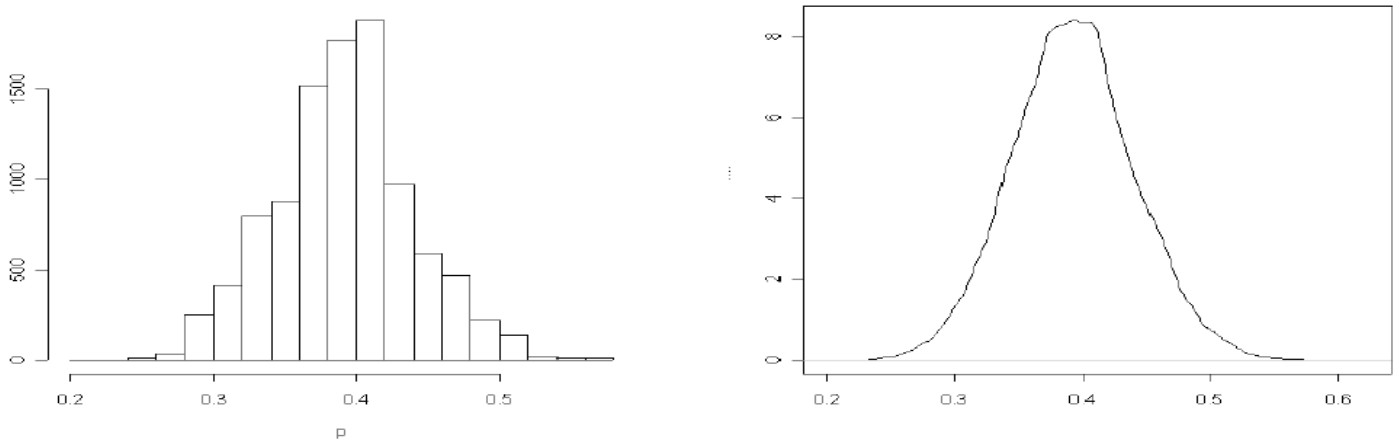
#### Résultats

L'estimateur du paramètre  $p$  obtenu avec une période de chauffe de 1000 sur 10000 itérations partant de  $p_0 = 1$  est :

$p$
0.3903357

Les représentations de ces résultats sont données dans les figures (4.13) et (4.14) :




 FIGURE 4.13 – Evolution de la chaîne du paramètre ( $p$ )

 FIGURE 4.14 – Histogramme et densité de la chaîne du paramètre ( $p$ )

### Etude de la convergence

Nous avons calculé le facteur d'échelle en utilisant  $M = 3$  et pour un nombre d'itération  $N = (1000, 5000, 10000)$ . Les résultats sont donnés dans le tableau suivant :

Nombre de chaînes	Nombre d'itérations	Facteur d'échelle
3	1000	1.04104
3	5000	1.01634
3	10000	1.0088

 TABLE 4.4 – Facteur d'échelle du paramètre ( $p$ )

Ces résultats montrent que le facteur d'échelle est proche de 1 d'où la bonne convergence à partir de 1000.

La convergence de l'estimateur de ( $p$ ) est donnée dans la figure suivante :

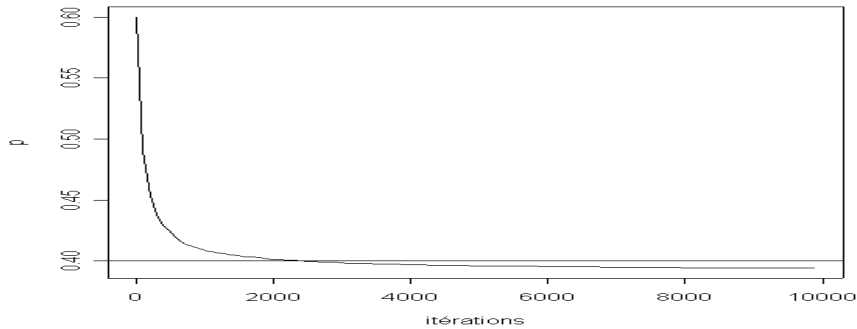


FIGURE 4.15 – Convergence de l'estimateur du paramètre ( $p$ )

#### 4.4.2 Méthode PMC

L'algorithme utilisé est celui de PMC général qui consiste à :

- Générer un vecteur initial  $p \sim \mathcal{U}(0, 1)$  de taille  $M$  donnée ;
- À l'étape  $t$  : générer un vecteur de taille  $M$  pour le paramètre  $p$  selon la loi  $\mathcal{U}(p^{(t-1)}, 1)$  (loi de proposition) et calculer le poids de chaque particule on utilisant la formule (4.1) ainsi les poids normalisés et rééchantillonné selon ces derniers .

#### Résultats

L'estimateur de  $p$  obtenu en générant 500 variables aléatoires pour  $T = 6$  est :

$p$
0.409745

Les représentations graphiques des résultats sont dans les figures (4.16) et (4.17) :

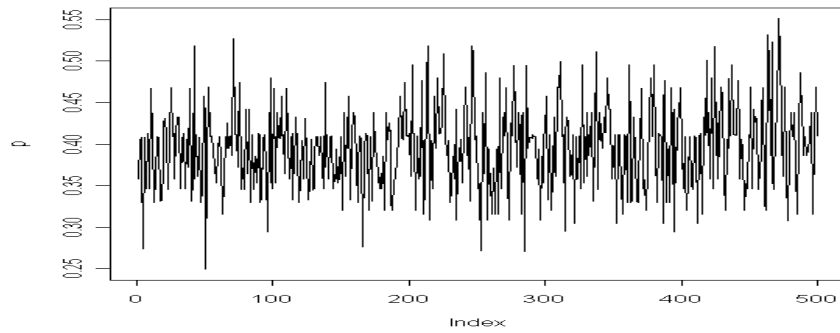


FIGURE 4.16 – Evolution de la chaîne du paramètre ( $p$ )

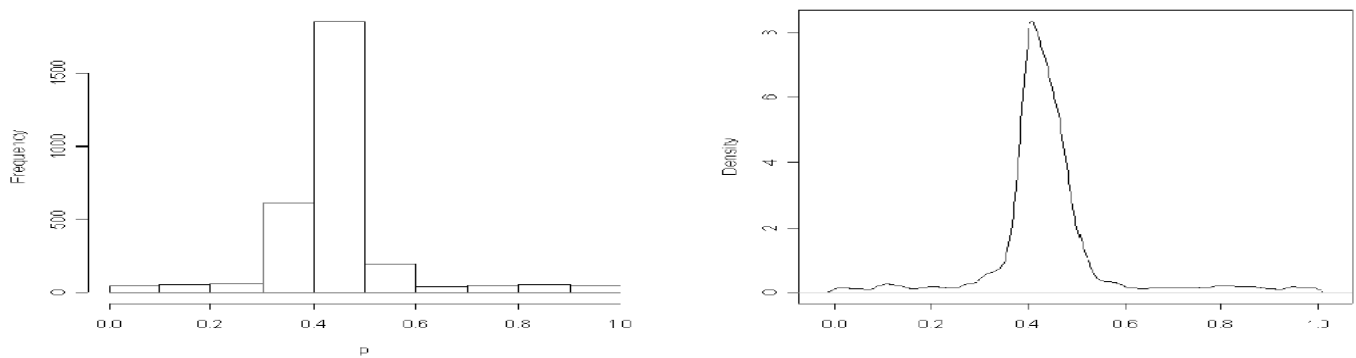


FIGURE 4.17 – Histogramme et densité de la chaîne du paramètre ( $p$ )

La convergence de l'estimateur de  $p$  est représentée dans la figure suivante :

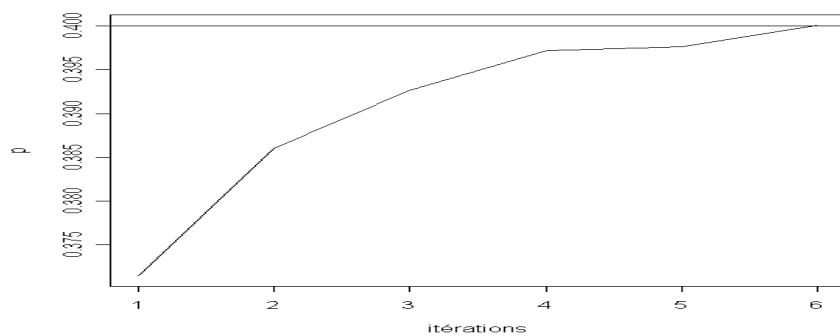


FIGURE 4.18 – Convergence de l'estimateur du paramètre ( $p$ )

### 4.4.3 Discussion des résultats

Pour ce modèle, les résultats montrent que pour la méthode MCMC, la chaîne de Markov obtenue converge vers la valeur exacte du paramètre  $p$  à partir d'un nombre d'itérations ( $> 2000$ ). Pour la méthode PMC, un nombre d'itération ( $T = 5$ ) est suffisant pour avoir une bonne approximation. Pour le temps d'exécution, MCMC fournit les résultats d'une durée de  $0.75s$  et PMC  $0.24s$ .

## 4.5 Modèle log Normale

### 4.5.1 Méthode MCMC

D'après les relations suivantes :

Une variable aléatoire  $X$  suit une loi log-normale quand son logarithme suit une loi normale ; c'est à dire que  $Y = \ln X$  suit une loi  $\mathcal{N}(\mu, \sigma)$ .

où  $\mu$  et  $\sigma$  sont les paramètres de la loi normale.

On a :

$$\mathbb{E}(X) = e^{\frac{\mu + \sigma^2}{2}};$$

$$\text{Var}(X) = (e^{\sigma^2} - 1)e^{2\mu + \sigma^2}.$$

Les estimateurs de la moyenne et de la variance obtenus en générant 400 variables aléatoires  $X \sim \text{Lnorm}(1.648721, 4.670774)$ , avec un nombre de période de chauffe est 3000 sur 10000 itérations sont :

$m$	$\sigma^2$
1.641440	4.6807272

Les représentations graphiques de ces résultats sont données dans les figures (4.19), (4.20), (4.21) et (4.22) :

La moyenne

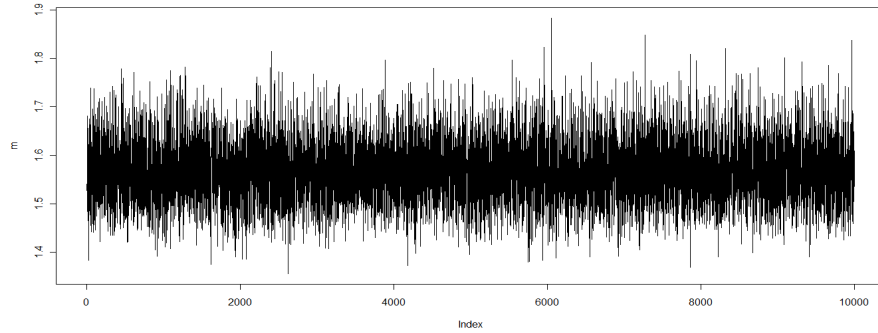


FIGURE 4.19 – Evolution de la chaîne de la moyenne ( $m$ )

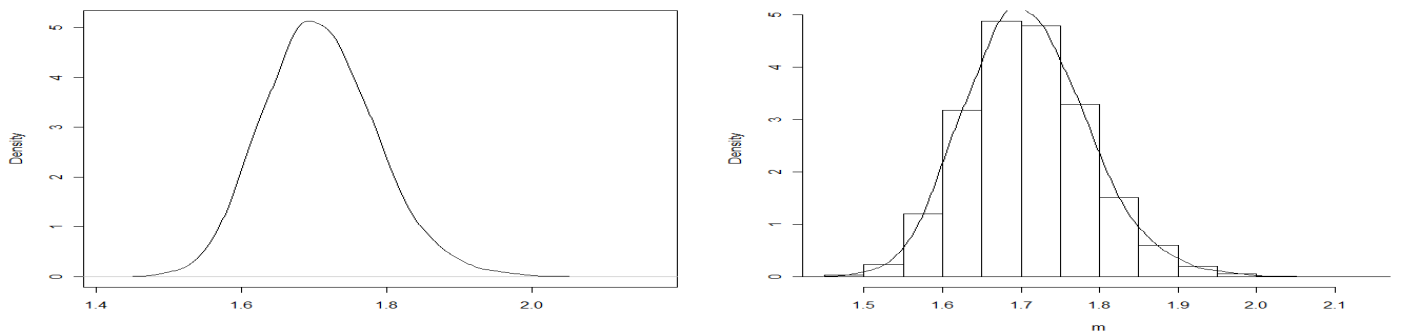


FIGURE 4.20 – Densité et histogramme et la chaîne de la moyenne ( $m$ )

La variance

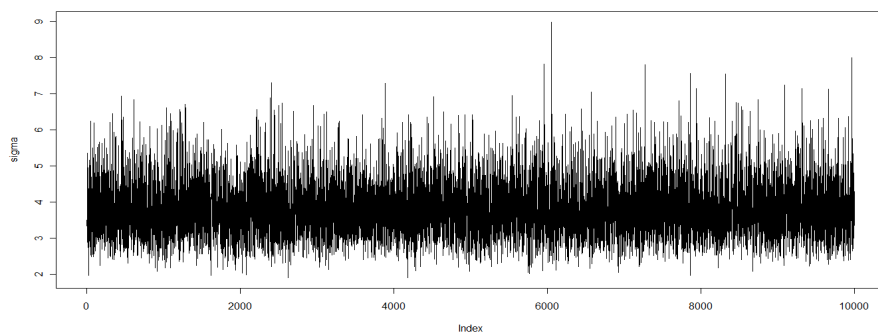


FIGURE 4.21 – Evolution de la chaîne de la variance ( $\sigma^2$ )

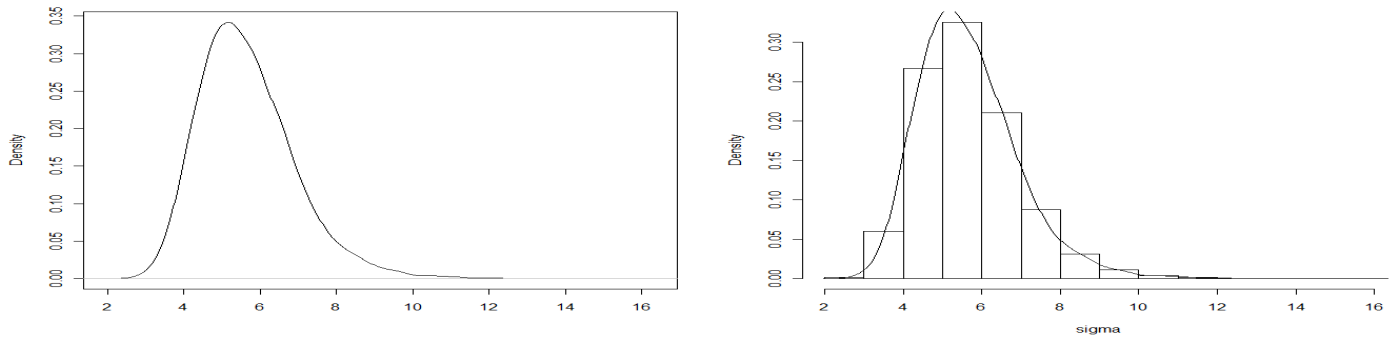


FIGURE 4.22 – Densité et histogramme et la chaîne de la variance ( $\sigma^2$ )

### Etude de la convergence

Pour vérifier la convergence, nous avons calculé le facteur d'échelle en utilisant  $M = 3$  et un nombre d'itérations  $N = (1000, 5000, 10000)$ .

Les résultats pour la moyenne sont donnés dans le tableau suivant :

Nombre de Chaînes	Nombre d'itérations	Facteur d'échelle
3	1000	1.093
3	5000	1.0374
3	10000	1.004

TABLE 4.5 – Facteur d'échelle de la moyenne ( $m$ )

La représentation graphique correspondante à la convergence de la moyenne est donnée comme suit :

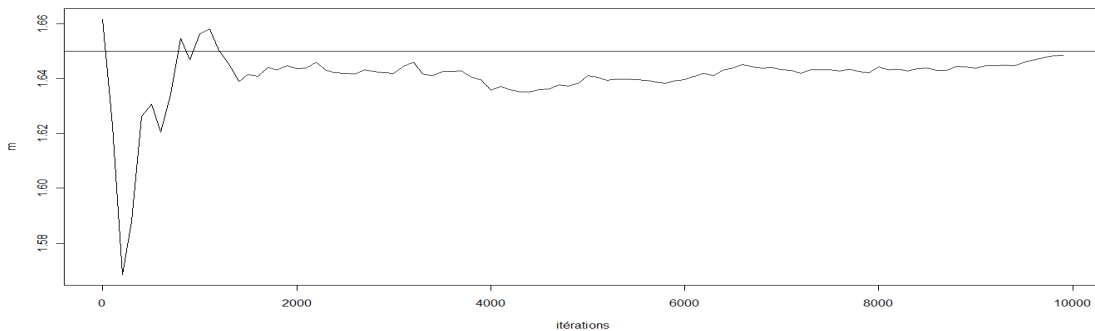


FIGURE 4.23 – Convergence de l'estimateur de la moyenne ( $m$ )

Les résultats pour la variance sont donnés dans le tableau suivant :

Nombre de chaînes	Nombre d'itérations	Facteur d'échelle
3	1000	1.051
3	5000	1.017
3	10000	1.0015

TABLE 4.6 – Facteur d'échelle de la variance ( $\sigma^2$ )

La représentation graphique correspondante à la convergence de la variance est donnée dans la figure suivante :

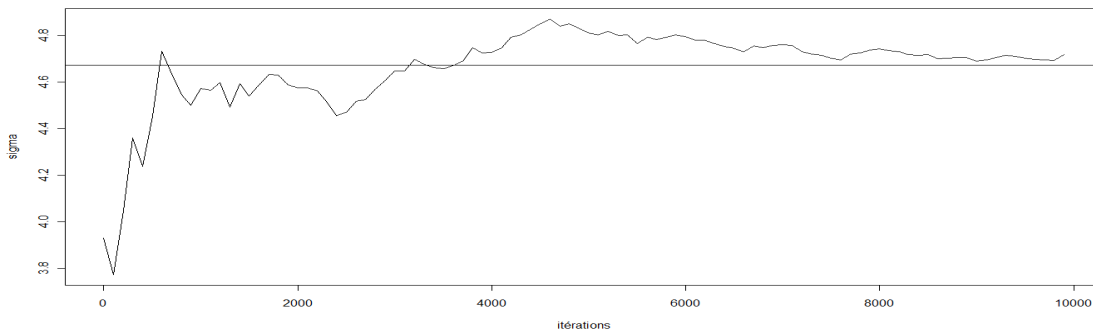


FIGURE 4.24 – Convergence de l'estimateur de variance ( $\sigma^2$ )

Ces résultats montrent que le facteur d'échelle est proche de 1 dans tous les cas. Pour un nombre d'itérations plus grands ( $> 5000$ ) le facteur donne une valeur exacte égale à 1 d'où la la bonne convergence.

### 4.5.2 Méthode PMC

Les estimateurs de la moyenne  $m$  et de la variance  $\sigma^2$  obtenus en générant 500 variables aléatoires  $X \sim Lnorm(1.648721, 4.670774)$ , pour  $T = 20$  sont :

$m$	$\sigma^2$
1.644047	4.67582

Les représentations graphiques des résultats sont données dans les figures (4.25), (4.26), (4.28) et (4.29) :

**La moyenne**

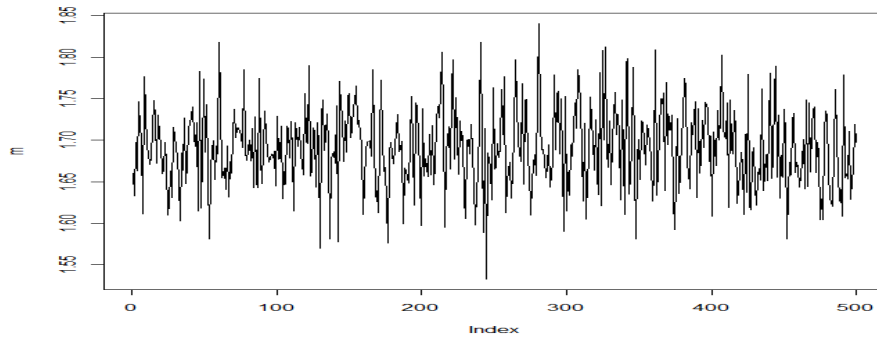


FIGURE 4.25 – Evolution de la chaîne de la moyenne ( $m$ )

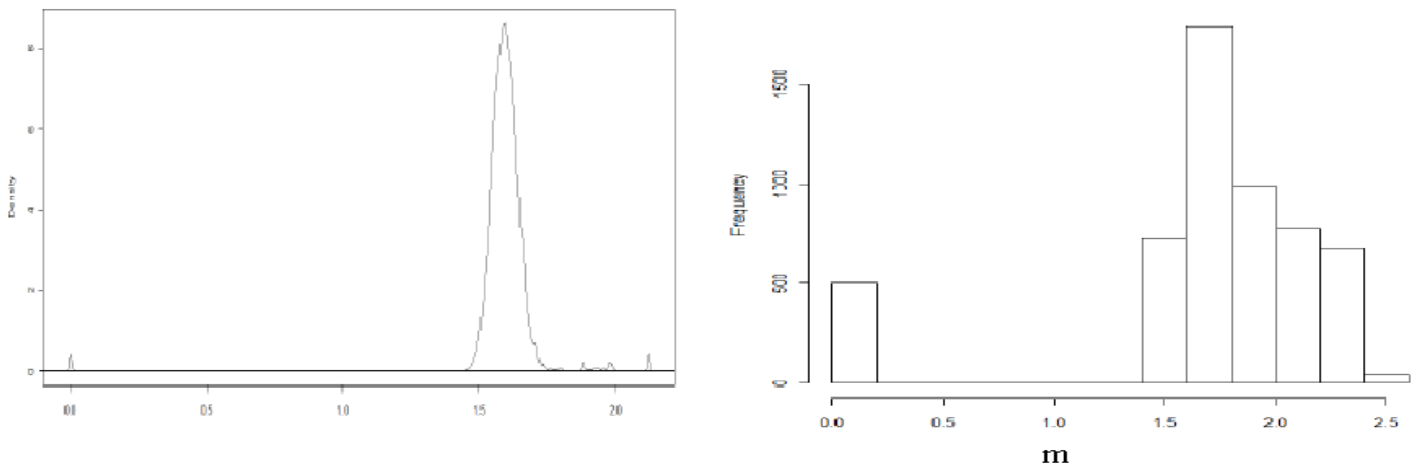


FIGURE 4.26 – Densité et histogramme de la chaîne de la moyenne ( $m$ )

La convergence de l'estimateur de la moyenne est donnée dans la figure suivante :



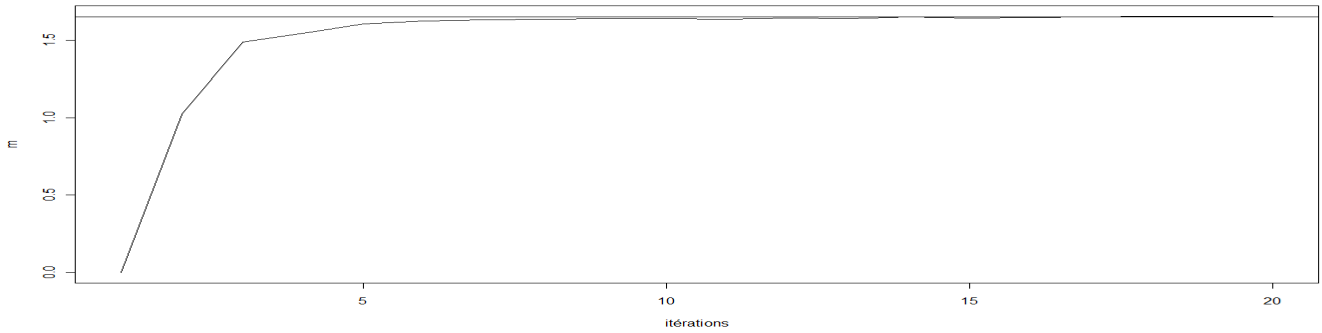


FIGURE 4.27 – Convergence de l'estimateur de la moyenne ( $m$ )

**La variance**

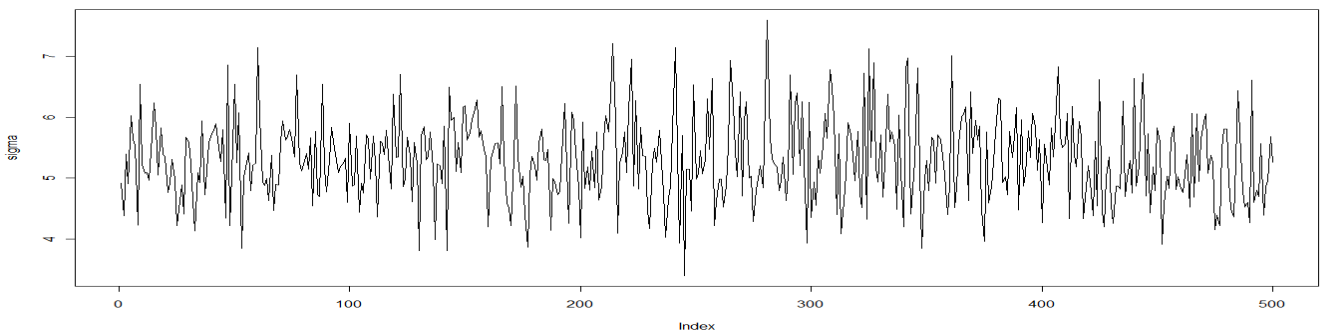


FIGURE 4.28 – Evolution de la chaîne de la variance ( $\sigma^2$ )

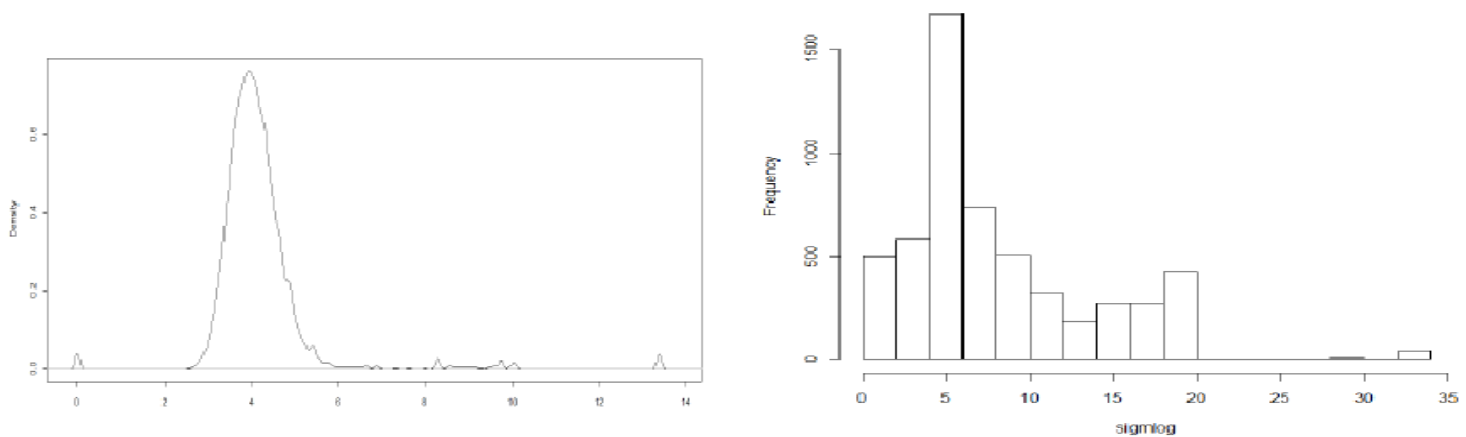


FIGURE 4.29 – Densité et histogramme de la chaîne de la variance ( $\sigma^2$ )

La convergence de l'estimateur de la variance est donnée dans la figure suivante :

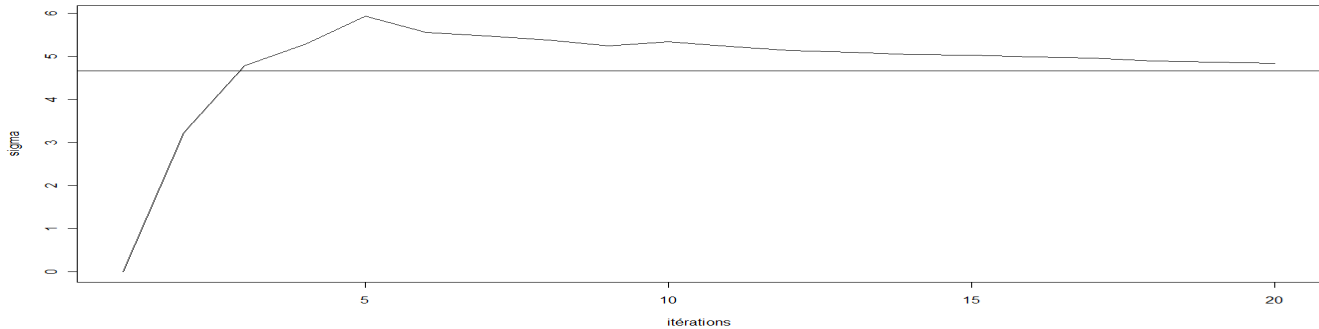


FIGURE 4.30 – Convergence de l'estimateur de la variance ( $\sigma^2$ )

### 4.5.3 Discussion des résultats

Les résultats montrent que pour la méthode MCMC, la chaîne de Markov obtenue pour la moyenne converge vers la valeur exacte du paramètre du modèle à partir d'un nombre d'itérations ( $> 6500$ ) et pour la variance à partir de ( $> 6000$ ) contrairement à la méthode PMC, pour un nombre d'itérations relativement réduit ( $T = 6$ ) pour la moyenne et ( $T = 15$ ) pour la variance cette méthode fournit d'excellentes approximations des deux paramètres.

Pour le temps d'exécution en fonction de la taille de l'échantillon on a les résultats du tableau qui suit :

	MCMC (Gibbs)			PMC		
La taille de l'échantillon	$m$	$\sigma^2$	<b>temps(s)</b>	$m$	$\sigma^2$	<b>temps(s)</b>
50	1.590338	4.615508	4.07	1.644047	4.703621	0.66
100	1.640744	4.689599	4.26	1.655545	4.667003	0.84
200	1.649146	4.670489	4.32	1.662685	4.689692	1.08
500	1.641440	4.687272	4.63	1.652368	4.686715	2.26

TABLE 4.7 – Temps d'exécution de MCMC et PMC en secondes

Ces résultats montrent que la méthode PMC fournit d'excellentes approximations pour une durée inférieure à celle de la méthode MCMC.

## 4.6 Modèle Poisson hiérarchique

Le modèle est défini comme suit [25] :

$$Y_i \sim \mathcal{P}(\theta); \quad i = 1, \dots, l;$$

$$Y_i \sim \mathcal{P}(\lambda); \quad i = l + 1, \dots, n;$$

où  $l$  est connu.

### 4.6.1 Méthode MCMC

L'algorithme utilisé est celui de Gibbs pour estimer les paramètres  $\theta$  et  $\lambda$ .

**La vraisemblance :**

$$L(\theta, \lambda | y) = \prod_{i=1}^l \frac{\theta^{y_i}}{(y_i)!} e^{-\theta} \times \prod_{i=l+1}^n \frac{\lambda^{y_i}}{(y_i)!} e^{-\lambda};$$

**Lois a priori :**

$$\theta \sim \mathcal{G}(a_1, b_1);$$

$$\lambda \sim \mathcal{G}(a_2, b_2);$$

**Lois conditionnelles :**

$$\theta | y, \lambda \sim \mathcal{G}(a_1 + \sum_{i=1}^l y_i, b_1 + l);$$

$$\lambda | y, \theta \sim \mathcal{G}(a_2 + \sum_{i=l+1}^n y_i, b_2 + n - l).$$

### Algorithme

- Initialiser les paramètres  $\theta_0, \lambda_0, a_1, a_2, b_1, b_2, n, l$  ;
- Générer un échantillon qui suit une loi de poisson de paramètre  $\theta_0$  de taille  $l$  et un autre de paramètre  $\lambda_0$  de taille  $(n - (l + 1))$  ;
- À l'étape  $t$ , tirer  $\theta$  à partir de la densité  $\mathcal{G}(a_1 + \sum_{i=1}^l y_i, b_1 + l)$  ;
- Tirer  $\lambda$  à partir de la densité  $\mathcal{G}(a_2 + \sum_{i=l+1}^n y_i, b_2 + n - l)$  ;
- Augmenter  $t$  à  $t + 1$ .

### Résultats

Les estimateurs de  $\theta$  et  $\lambda$  obtenus avec  $\theta_0 = 1$  et  $\lambda_0 = 3$ , sur 10000 itérations et la période de chauffe est 3000 sont :

$\theta$	$\lambda$
1.060824	3.008664

Les représentations graphiques de ces résultats sont données dans les figures (4.31), (4.32), (4.33) et (4.34) :

#### Paramètre $\theta$

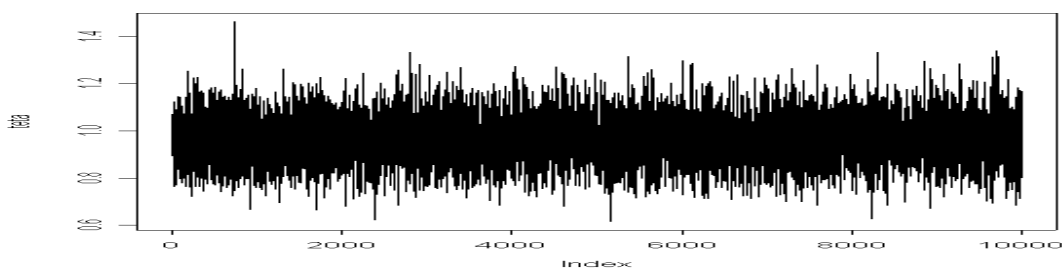


FIGURE 4.31 – Evolution de la chaîne du paramètre ( $\theta$ )

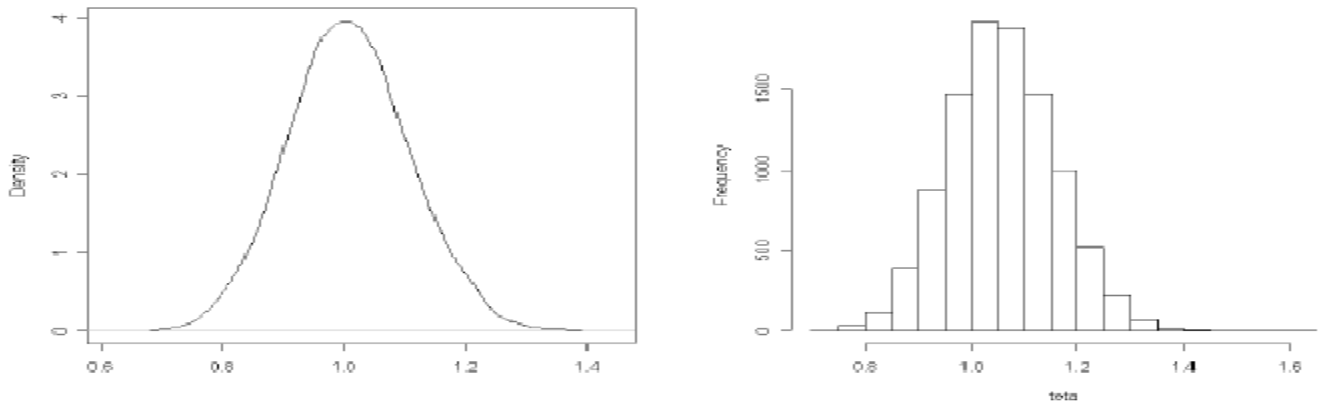


FIGURE 4.32 – Densité et histogramme de la chaîne du paramètre ( $\theta$ )

**Paramètre  $\lambda$**

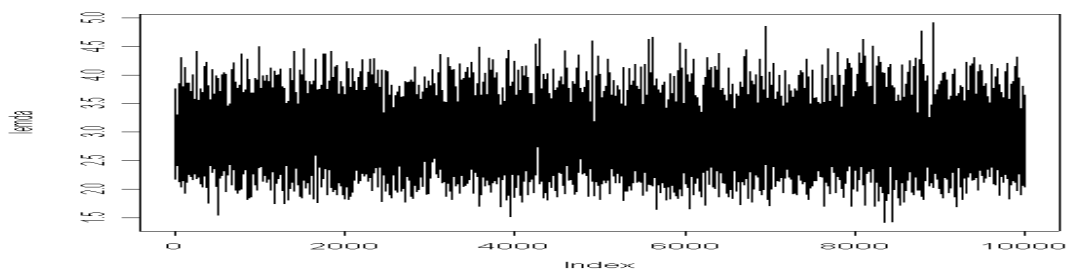


FIGURE 4.33 – Evolution de la chaîne du paramètre ( $\lambda$ )

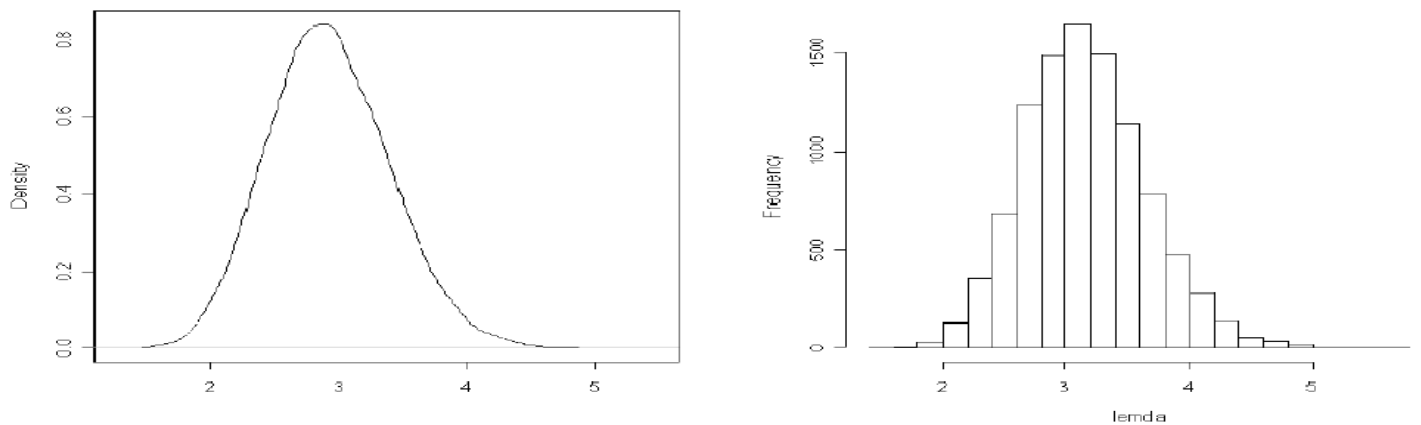


FIGURE 4.34 – Densité et histogramme de la chaîne du paramètre ( $\lambda$ )

### Etude de la convergence

Pour vérifier la convergence, nous avons calculé le facteur d'échelle en utilisant  $M = 3$  et pour un nombre d'itérations  $N = (1000, 5000, 10000)$ .

Les résultats pour le paramètre  $\theta$  sont donnés dans le tableau suivant :

Nombre de Chaînes	Nombre d'itérations	Facteur d'échelle
3	1000	1.3678
3	5000	1.01487
3	10000	1.0039

TABLE 4.8 – Facteur d'échelle du paramètre  $\theta$

La convergence de l'estimateur du paramètre  $\theta$  est dans la figure suivante :

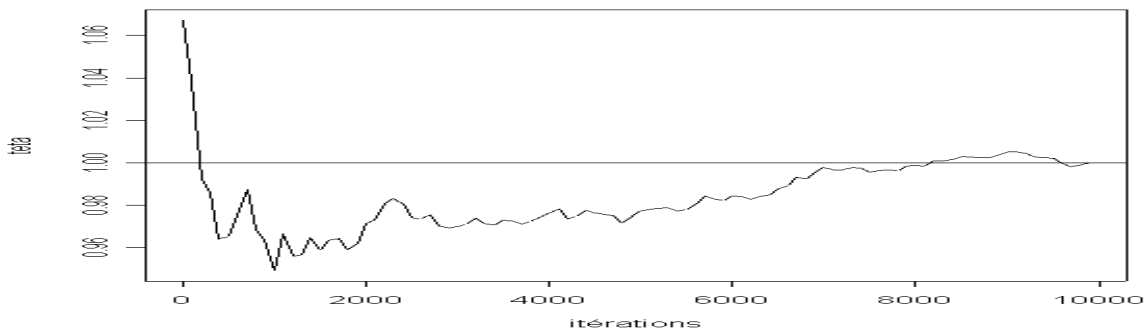


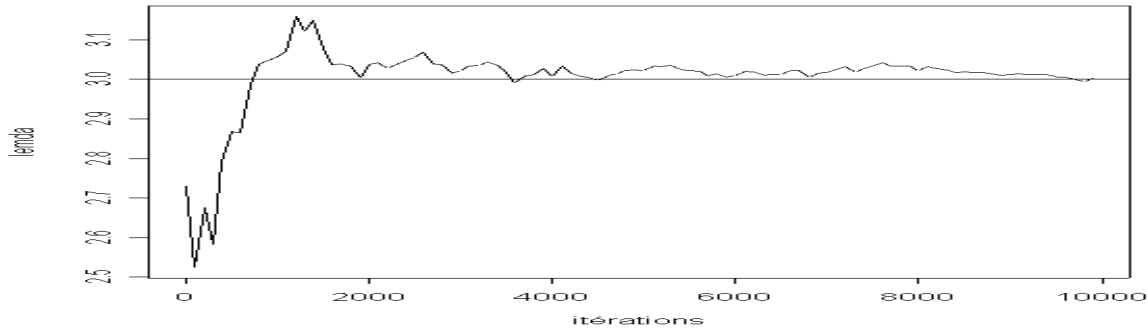
FIGURE 4.35 – Convergence de l'estimateur du paramètre ( $\theta$ )

Les résultats pour le paramètre  $\lambda$  sont donnés dans le tableau suivant :

Nombre chaînes	Nombre d'itérations	Facteur d'échelle
3	1000	1.15678
3	5000	1.0628
3	10000	1.00721

TABLE 4.9 – Facteur d'échelle du paramètre  $\lambda$

La représentation graphique de la convergence de l'estimateur est donnée comme suit :


 FIGURE 4.36 – Convergence de l'estimateur du paramètre ( $\lambda$ )

### 4.6.2 Méthode PMC

L'algorithme utilisé est celui de PMC général qui consiste à :

- Initialiser les paramètres  $\theta_0, \lambda_0, a_1, a_2, b_1, b_2, n, l$  ;
- Générer un échantillon suit une loi de poisson de paramètre  $\theta_0$  de taille  $l$  et un autre de paramètre  $\lambda_0$  de taille  $(n - (l + 1))$  ;
- Générer un vecteur initial  $\theta \sim \mathcal{U}_{[0,1]}$  de taille  $M$  donnée ;
- Générer un vecteur initial  $\lambda \sim \mathcal{U}_{[0,1]}$  de taille  $M$  ;
- Les lois de propositions sont des lois *Normale* tranquées sur  $[0, \infty[$  pour les deux paramètres ;
- À l'étape  $t$  : générer un vecteur de taille  $M$  pour le paramètre  $\theta$  ainsi pour  $\lambda$  selon les lois de propositions , calculer le poids de chaque particule ainsi que les poids normalisés et rééchantillonner selon ces derniers .

Les estimateurs de  $\theta$  et de  $\lambda$  obtenus pour  $T = 20$  sont :

$\theta$	$\lambda$
1.051303	3.001309

Les représentations graphiques des résultats sont dans les figures (4.37), (4.38), (4.40) et (4.41) :

**Paramètre  $\theta$**

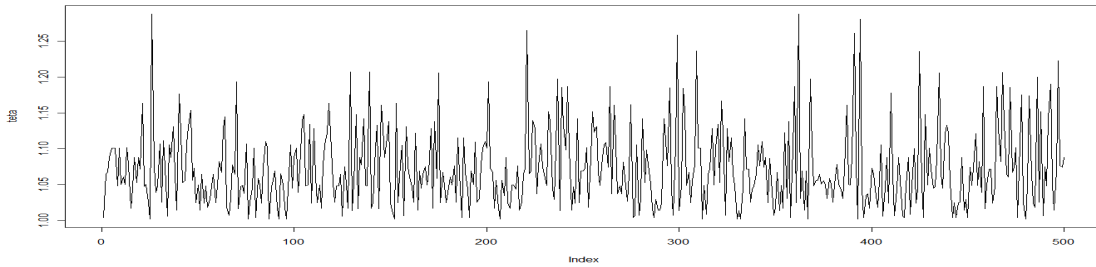


FIGURE 4.37 – Evolution de la chaîne du paramètre ( $\theta$ )

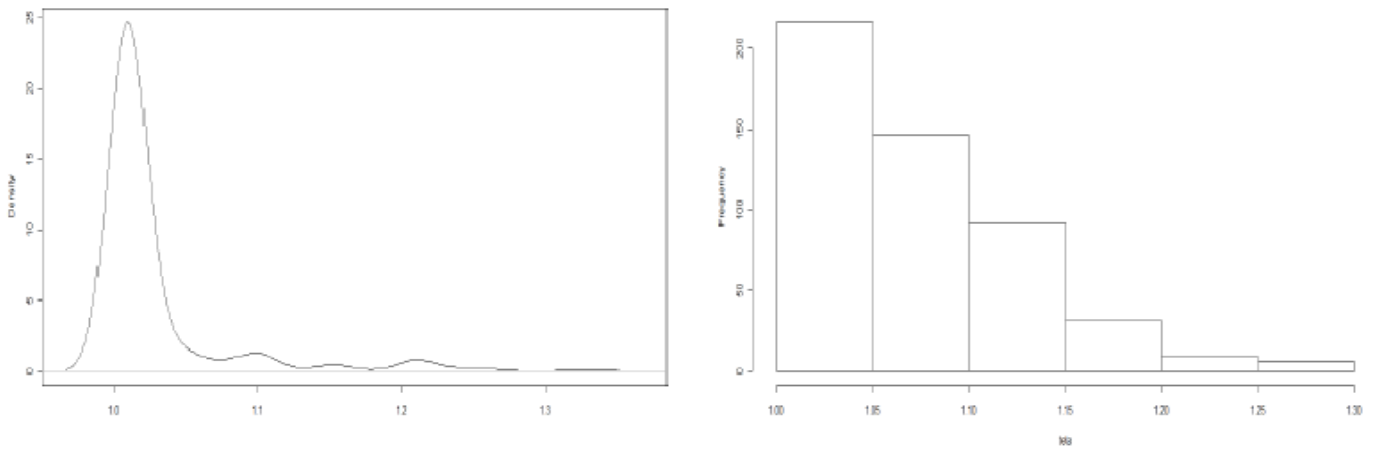


FIGURE 4.38 – Densité et histogramme du paramètre ( $\theta$ )

La convergence de l'estimateur du paramètre  $\theta$  est donnée comme suit :



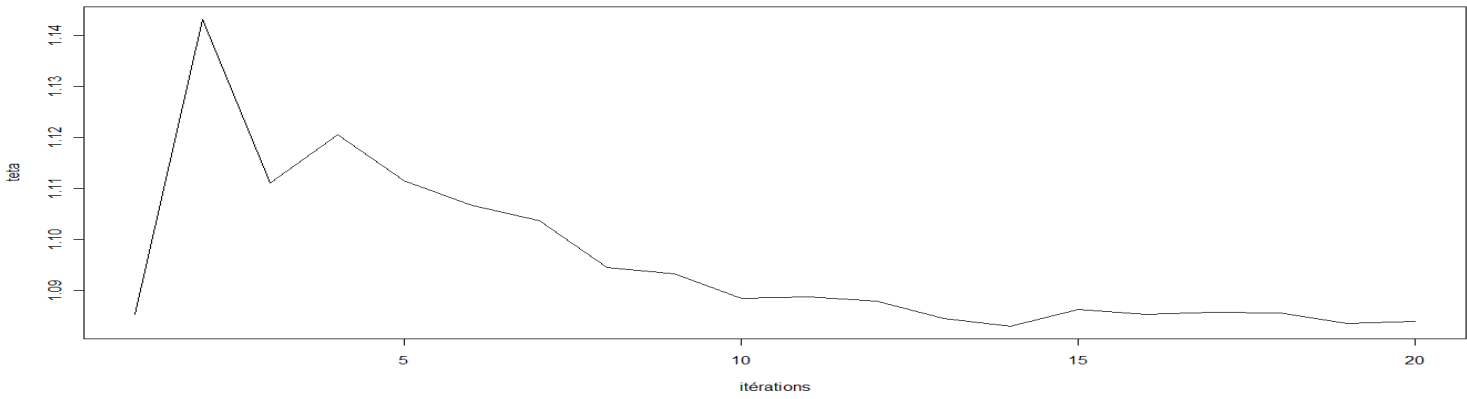


FIGURE 4.39 – Convergence de l’estimateur du paramètre ( $\theta$ )

Paramètre  $\lambda$

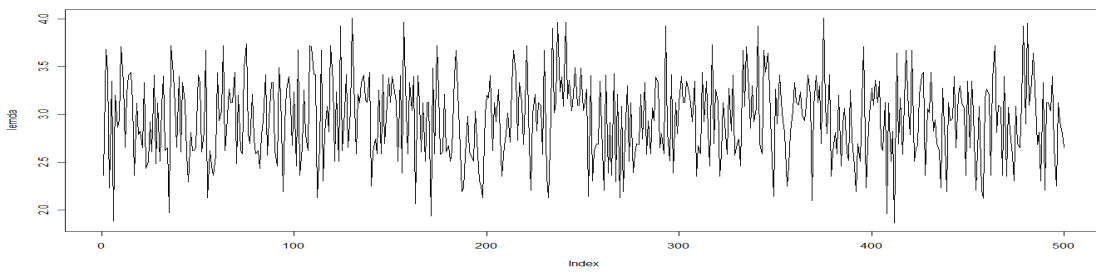


FIGURE 4.40 – Evolution de la chaîne du paramètre ( $\lambda$ )

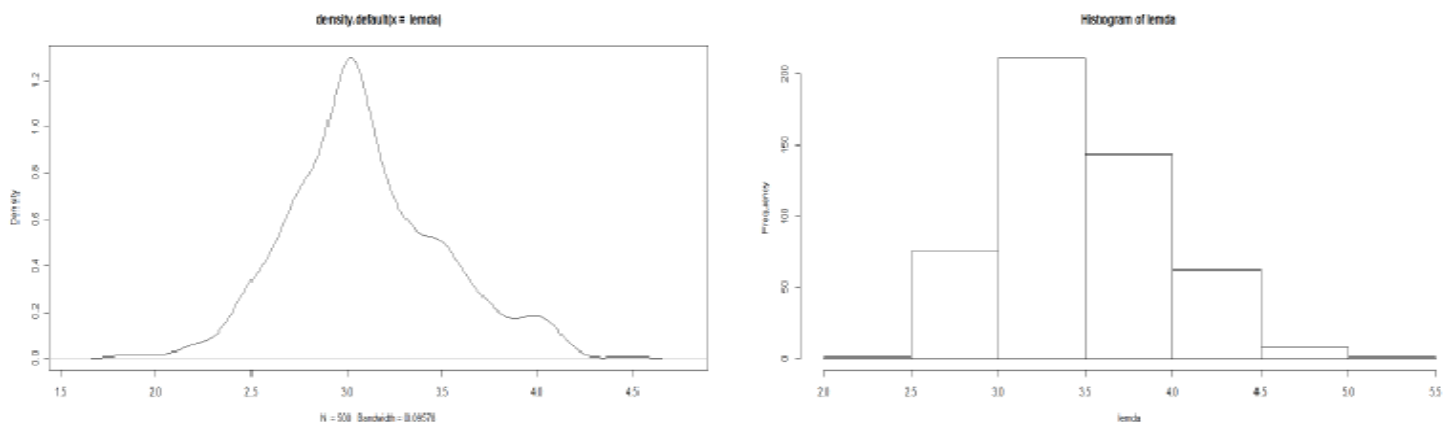


FIGURE 4.41 – Densité et histogramme du paramètre ( $\lambda$ )

La convergence de l'estimateur du paramètre  $\lambda$  est donnée comme suit :

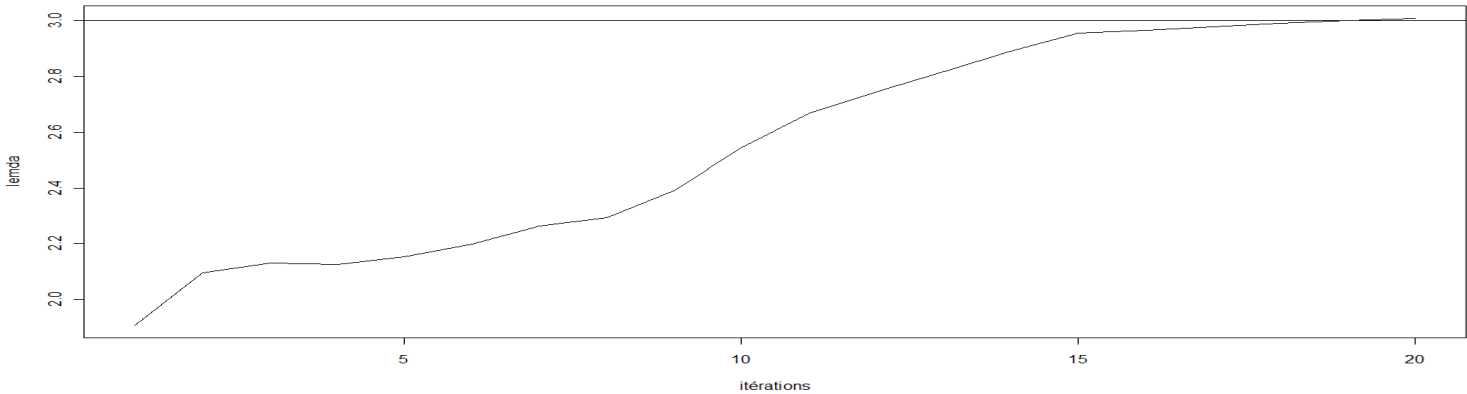


FIGURE 4.42 – Convergence de l'estimateur du paramètre ( $\lambda$ )

### 4.6.3 Discussion des résultats

Les résultats montrent que pour la méthode MCMC, la chaîne de Markov obtenue pour le paramètre  $\theta$  converge vers la valeur exacte du paramètre du modèle à partir d'un nombre d'itérations ( $> 7000$ ) et pour  $\lambda$  à partir de ( $> 2000$ ) contrairement à la méthode PMC, pour un nombre d'itérations réduit ( $T = 15$ ) pour les paramètres  $\theta$  et  $\lambda$ . Cette méthode fournit d'excellentes approximations des deux paramètres.

Pour le temps d'exécution, MCMC fournit les résultats pour une durée de 1.53s et pour PMC une durée 0.79s.

## 4.7 Modèle de Bernoulli

Soit  $X_1, X_2, \dots, X_n \sim \mathcal{B}(p)$  alors :

$$f(x, p) = p^x (1 - p)^{1-x}.$$

### 4.7.1 Méthode MCMC

La vraisemblance :

$$L(p|x) = p^{\sum_{i=1}^n x_i} (1 - p)^{n - \sum_{i=1}^n x_i}.$$

Loi a priori du paramètre :

$$p \sim \text{Beta}(\alpha, \beta).$$

Loi conditionnelle :

$$p|x \sim \text{Beta}\left(\alpha + \sum_{i=1}^n x_i, \beta + n - \sum_{i=1}^n x_i\right).$$

Puisque on a la loi conditionnelle de  $p$  alors il est facile d'appliquer l'échantionneur de Gibbs.

### Algorithme

L'algorithme utilisé est celui de Gibbs, sur un échantillon  $X$  généré selon la loi de Bernoulli de paramètre  $p = 0.5$  de taille 500, le nombre d'itérations est 10000 et la période de chauffe est 1000 (Voir l'annexe)

### Résultats

L'estimateur du paramètre  $p$  obtenu est

$p$
0.5091636

Les représentations graphiques des résultats sont dans les figures (4.43) et (4.44) :

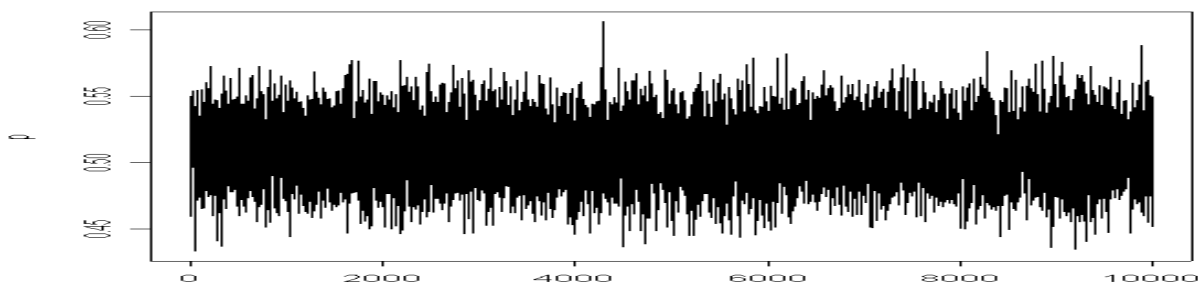
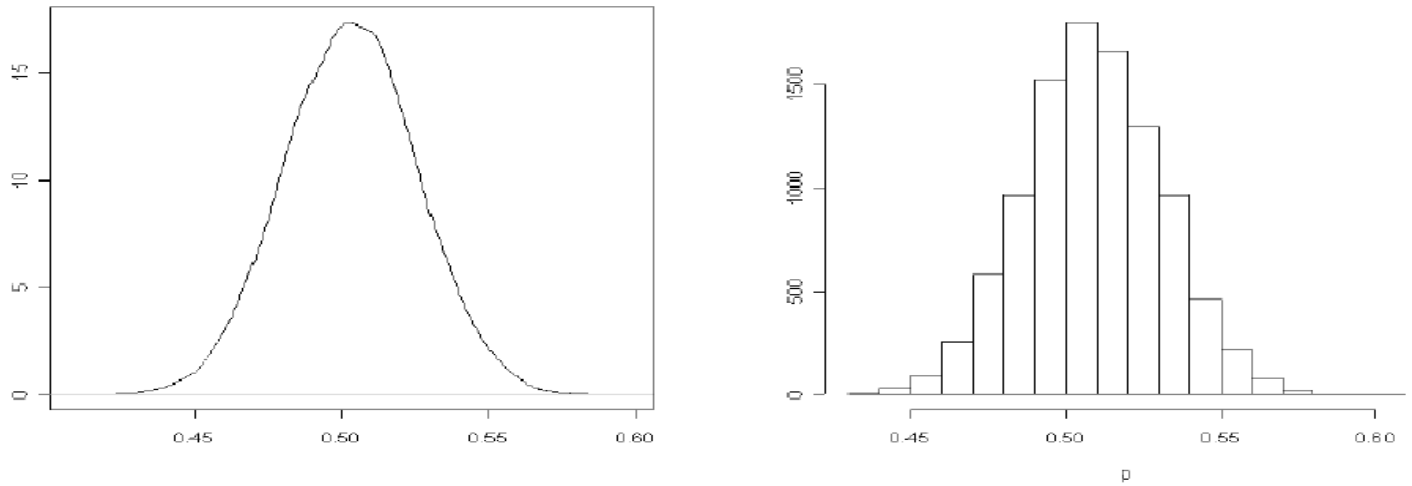


FIGURE 4.43 – Evolution de la chaîne du paramètre ( $p$ )


 FIGURE 4.44 – Densité et histogramme du paramètre ( $p$ )

### Etude de la convergence

Pour vérifier la convergence, nous avons calculé le facteur d'échelle en utilisant  $M = 3$  et pour le nombre d'itérations  $N = (1000, 5000, 10000)$ . Les résultats pour le paramètre  $p$  sont donnés dans le tableau suivant :

Nombre de Chaînes	Nombre d'itérations	Facteur d'échelle
3	1000	1.26
3	5000	1.1245
3	10000	1.05

 TABLE 4.10 – Facteur d'échelle du paramètre ( $p$ )

La convergence de l'estimateur du paramètre  $p$  est dans la figure suivante :

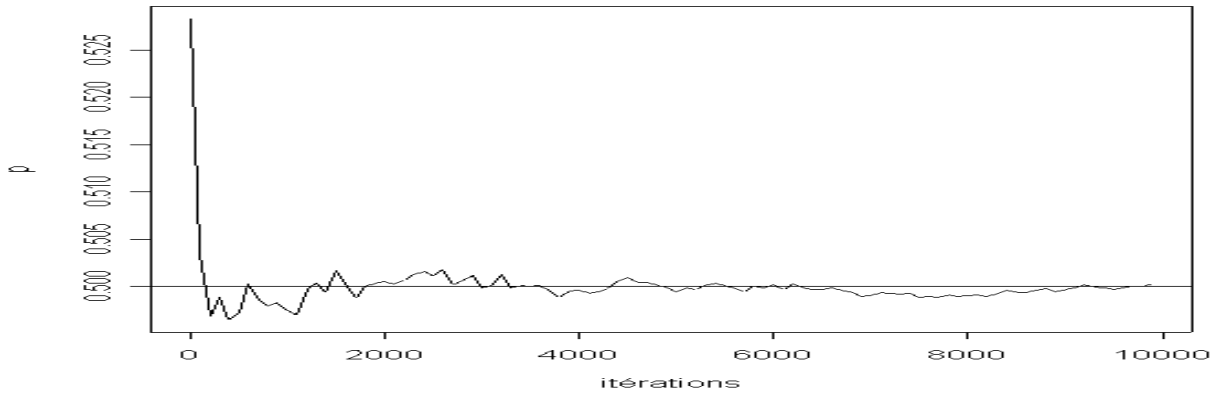


FIGURE 4.45 – Convergence de l’estimateur du paramètre  $p$

### 4.7.2 Méthode PMC

L’algorithme utilisé est celui de PMC général (Voir L’annexe).

#### Résultats

L’estimateur de  $p$  obtenu pour  $T = 10$  est :

$p$
0.503716

Les représentations graphiques de ces résultats sont dans les figures (4.46) et (4.47) :

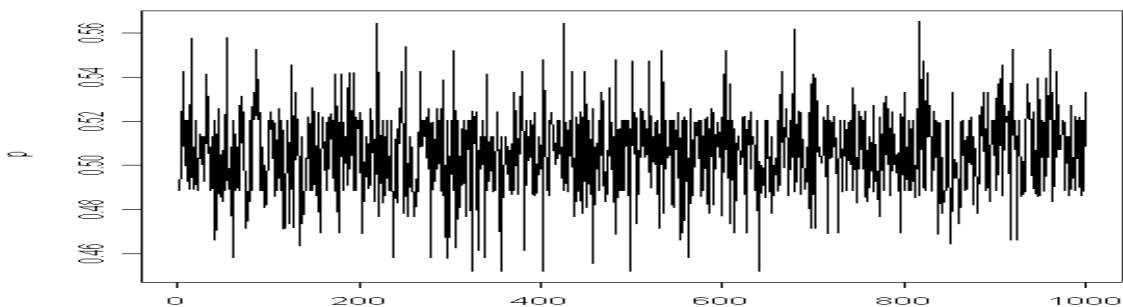


FIGURE 4.46 – Evolution de la chaîne du paramètre ( $p$ )

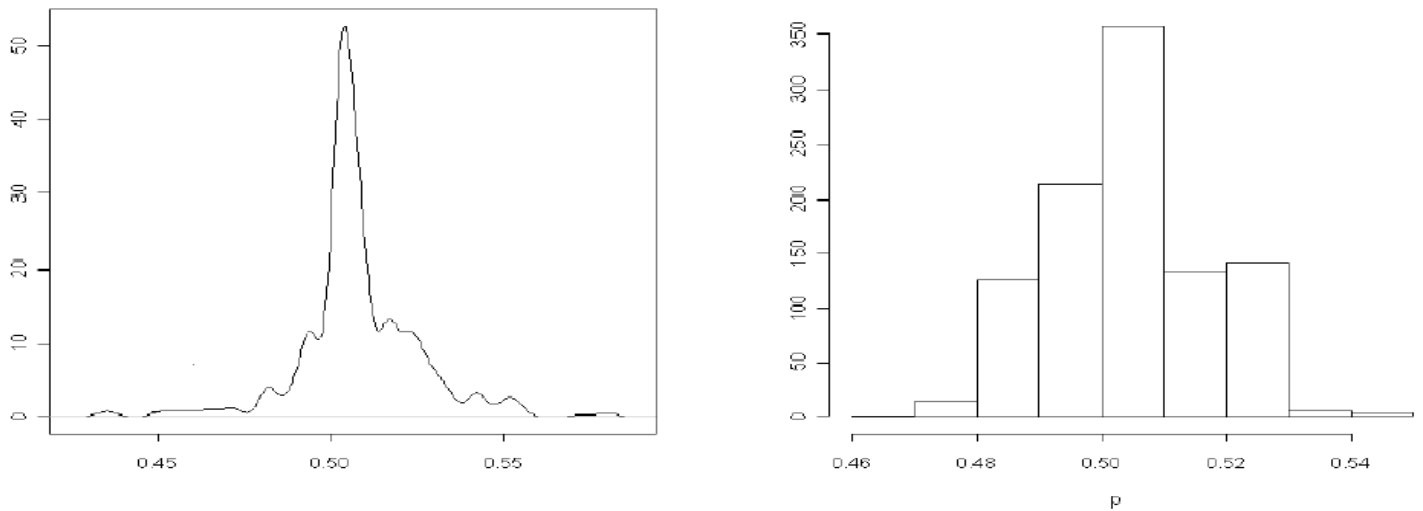


FIGURE 4.47 – Densité et histogramme du paramètre ( $p$ )

La convergence de l'estimateur du paramètre  $p$  est donnée comme suit :

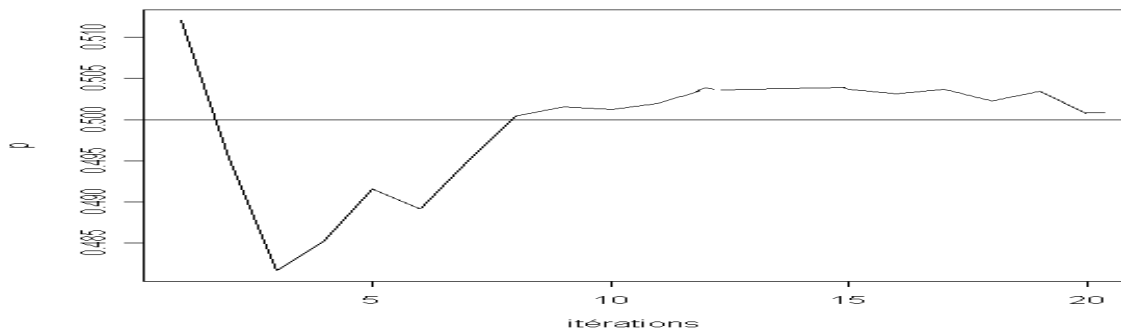


FIGURE 4.48 – Convergence de l'estimateur du paramètre ( $p$ )

### 4.7.3 Discussion des résultats

Les résultats montrent que pour la méthode MCMC, la chaîne de Markov obtenue pour le paramètre  $p$  converge vers la valeur exacte du paramètre du modèle à partir d'un nombre d'itérations ( $> 2000$ ) contrairement à la méthode PMC, pour un nombre d'itérations réduit ( $T = 8$ ). Cette méthode fournit d'excellentes approximations du paramètre.

Pour le temps d'exécution, MCMC fournit les résultats pour une durée 0.47s et pour PMC une durée de 0.33s.

## 4.8 Modèle Exponentiel à données manquantes

Soit  $n$  données représentent les temps de panne d'une machine qui sont censurées (incomplètes)[8]. Ces données sont simulées selon une loi *exponentielle* de paramètre  $\theta = 0.8519$ ; avec un censure de  $c = 0.4$ , c'est-à-dire 40%.

### Loi a priori

$$\theta \sim \mathcal{G}(a, b).$$

### Lois conditionnelles :

$$\theta|y, z \sim \mathcal{G}(n + a, s + r * c + z + b);$$

$$z|y, \theta \sim \mathcal{G}(r, \theta);$$

### 4.8.1 Méthode MCMC

#### Algorithme

L'algorithme utilisé est celui de Gibbs, sur un échantillon  $X$  généré selon la loi exponentielle de paramètre  $\theta = 0.8519$  de taille 20, le nombre d'itérations est 10000 et la période de chauffe est 3000, (Voir l'annexe).

L'estimateur du paramètre  $\theta$  obtenu est :

$\theta$
0.8545777

Les représentations graphiques des résultats sont dans les figures (4.49) et (4.50) :

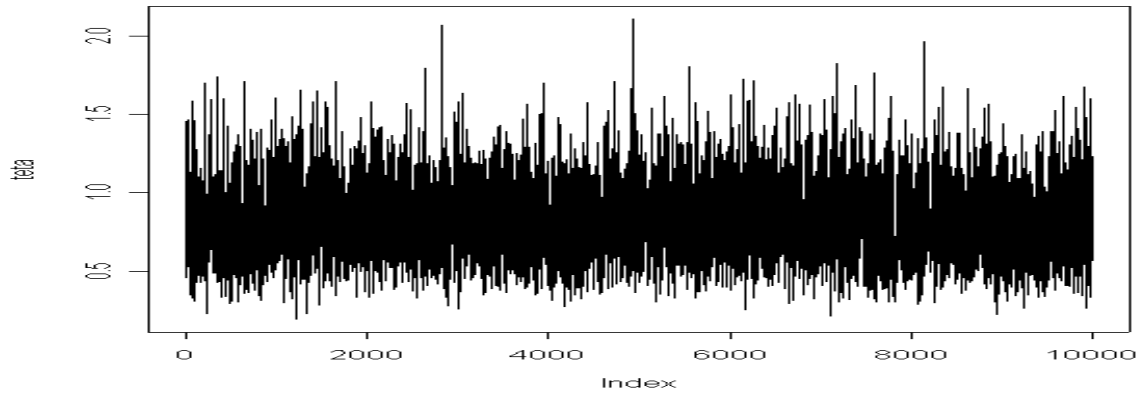


FIGURE 4.49 – Evolution de la chaîne du paramètre ( $\theta$ )

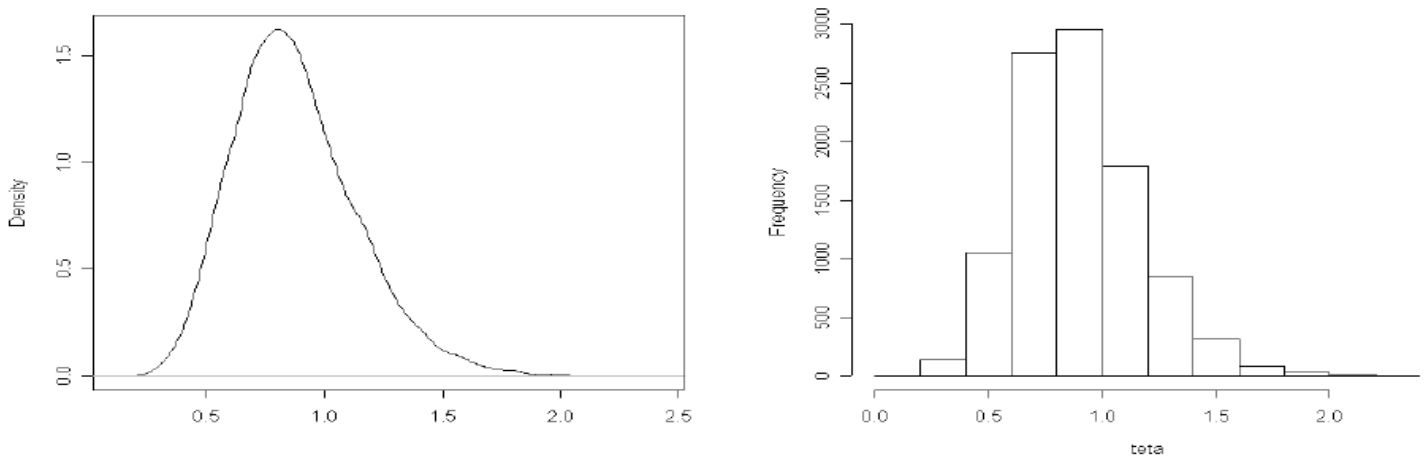


FIGURE 4.50 – Densité et histogramme de la chaîne du paramètre ( $\theta$ )

La convergence de l'estimateur du paramètre  $\theta$  est donnée comme suit :

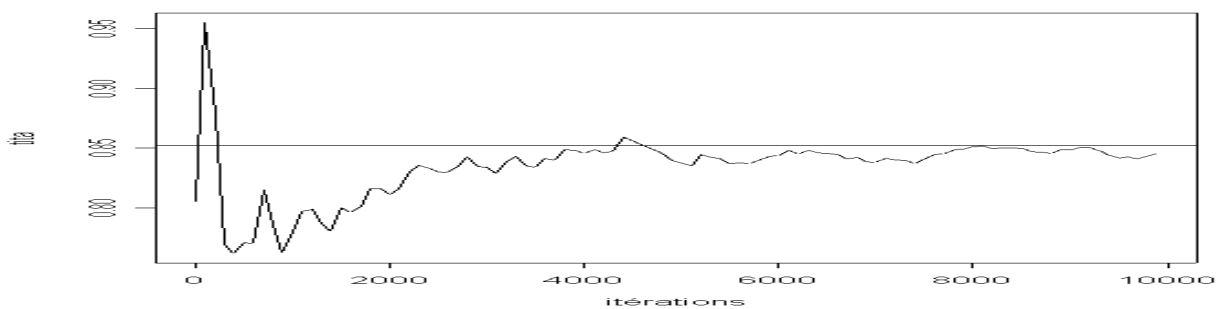


FIGURE 4.51 – Convergence de l'estimateur du paramètre ( $\theta$ )



## 4.8.2 Méthode PMC

### Algorithme

L'algorithme utilisé est celui de Rao–Blackwell à données manquantes (Voir L'annexe).

### Résultats

L'estimateur de  $\theta$  obtenu pour  $T = 30$  est :

$\theta$
0.8514

Les représentations graphiques de ces résultats sont dans les figures (4.52) et (4.53) :

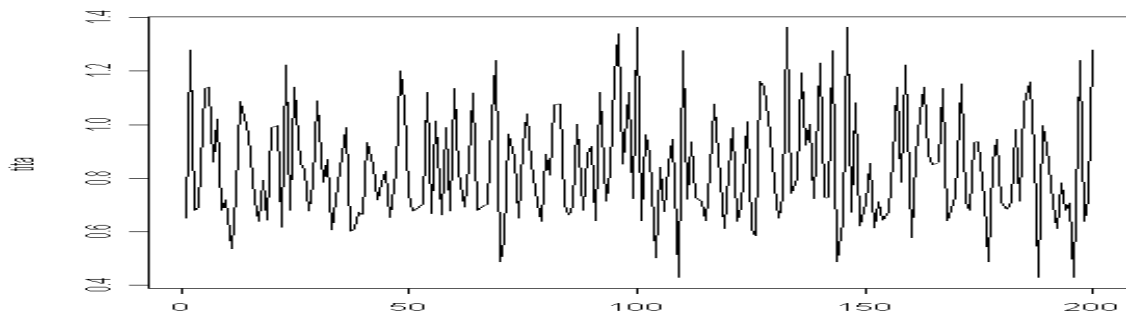


FIGURE 4.52 – Evolution de la chaîne du paramètre ( $\theta$ )

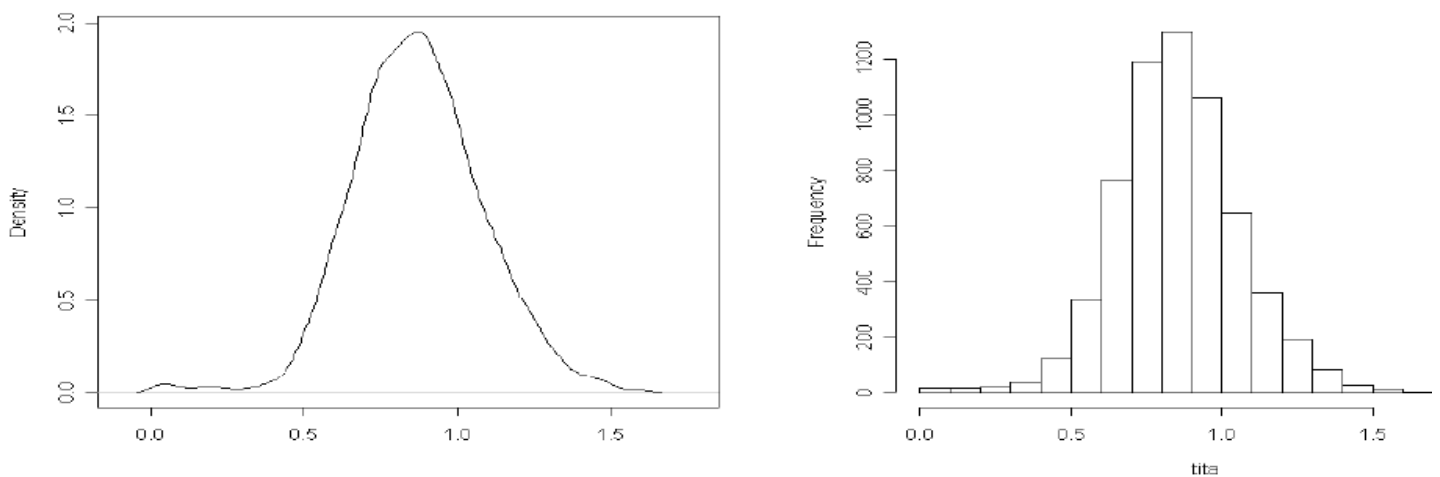


FIGURE 4.53 – Densité et histogramme de de la chaîne du paramètre ( $\theta$ )

La convergence de l'estimateur du paramètre  $\theta$  est donnée comme suit :

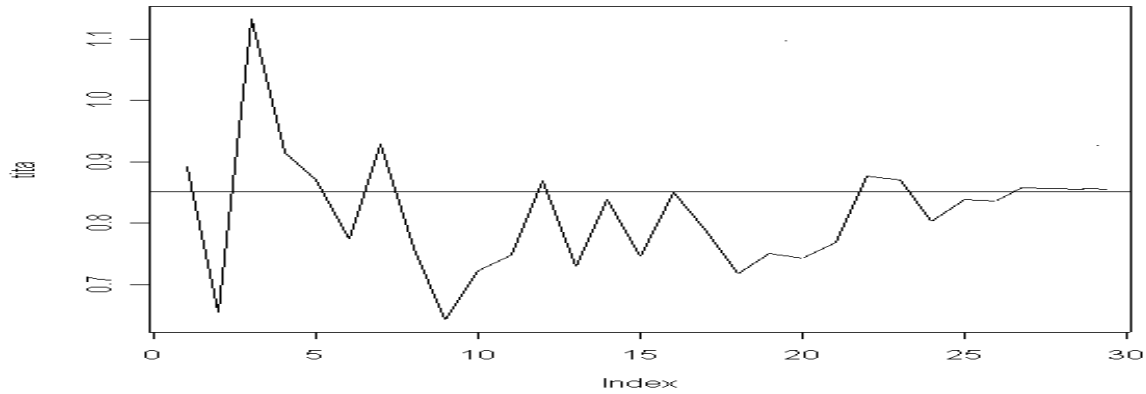


FIGURE 4.54 – Convergence de l'estimateur du paramètre ( $\theta$ )

### 4.8.3 Discussion des résultats

Les résultats montrent que pour la méthode MCMC, la chaîne de Markov obtenue pour le paramètre  $\theta$  converge vers la valeur exacte du paramètre du modèle à partir d'un nombre d'itérations ( $>4000$ ) contrairement à la méthode PMC, pour un nombre d'itérations relativement réduit ( $T=25$ ). Cette méthode fournit d'excellentes approximations du paramètre estimé.

Par contre pour le temps d'exécution, MCMC fournit les résultats d'une durée de 1.42s et PMC 58.58s ce qui prouve l'inconvénient de Rao-Blackwell.

## 4.9 Partie 2 : Application réelle

### 4.10 Modèle à changement de point

**Définition 5.** Soit  $Y_n = \{y_1, y_2 \dots y_n\}$  une série chronologique tel que  $Y_n \sim A(\alpha)$  ( $A$  est un modèle statistique quelconque (Poisson, Bernoulli...)) où la valeur du paramètre  $\alpha$  change à un temps inconnu

$T_m = \{t_1, t_2 \dots t_m\}$  [27],

$$\alpha = \begin{cases} \alpha_1 & t \leq t_1 \\ \alpha_2 & t_1 < t \leq t_2 \\ \cdot & \\ \cdot & \\ \cdot & \\ \alpha_m & t_{m-1} < t \leq t_m \end{cases}$$

Le modèle  $A$  est dit à changement de point.

## 4.11 Application et résultats

Les données analysées sont obtenues par Jarret (1979) et elles concernent le nombre de catastrophes causés par le charbonnage aux niveau des mines de la grande-Bretagne durant une période de 112 années (1851-1962) ; les observations annuelles du nombre de catastrophes est mentionnées dans le tableau suivant[7, 17] :

Year	C	Year	C	Year	C	Year	C	Year	C	Year	C
1851	4	1871	5	1891	2	1911	0	1931	3	1951	1
1852	5	1872	3	1892	1	1912	1	1932	3	1952	0
1853	4	1873	1	1893	1	1913	1	1933	1	1953	0
1854	1	1874	4	1894	1	1914	1	1934	1	1954	0
1855	0	1875	4	1895	1	1915	0	1935	2	1955	0
1856	4	1876	1	1896	3	1916	1	1936	1	1956	0
1857	3	1877	5	1897	0	1917	0	1937	1	1957	1
1858	4	1878	5	1898	0	1918	1	1938	1	1958	0
1859	0	1879	3	1899	1	1919	0	1939	1	1959	0
1860	6	1880	4	1900	0	1920	0	1940	2	1960	1
1861	3	1881	2	1901	1	1921	0	1941	4	1961	0
1862	3	1882	5	1902	1	1922	2	1942	2	1962	1
1863	4	1883	2	1903	0	1923	1	1943	0		
1864	0	1884	2	1904	0	1924	0	1944	0		
1865	2	1885	3	1905	3	1925	0	1945	0		
1866	6	1886	4	1906	1	1926	0	1946	1		
1867	3	1887	2	1907	0	1927	1	1947	4		
1868	3	1888	1	1908	3	1928	1	1948	0		
1869	5	1889	3	1909	2	1929	0	1949	0		
1870	4	1890	2	1910	2	1930	2	1950	0		

FIGURE 4.55 – Nombre de catastrophes par an causés par le charbonnage en Bretagne

La représentation graphique des données est illustrées dans la figure suivante :

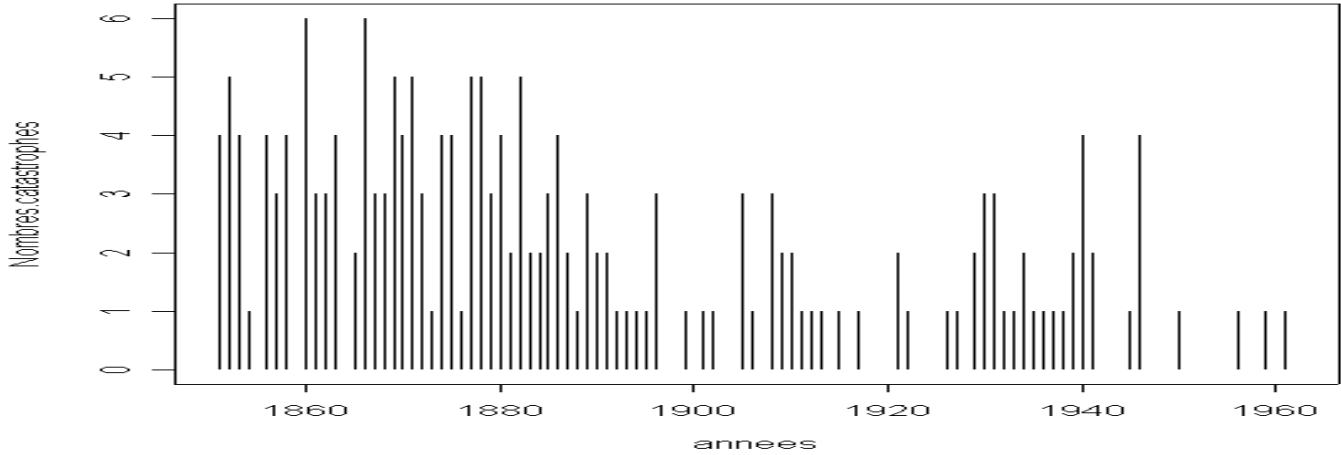


FIGURE 4.56 – Nombre de catastrophes

Ces données montrent qu'il y a une diminution du nombre de catastrophes durant cette période; le modèle proposé par Carlin et al.(1992) est :

$$Y_i \sim \mathcal{P}(\theta); \quad i = 1, \dots, k;$$

$$Y_i \sim \mathcal{P}(\lambda); \quad i = k + 1, \dots, 112.$$

Durant la période  $k$  la distribution est poissonnienne de paramètre  $\theta$ , au delà de  $k$  c'est poissonnienne de paramètre  $\lambda$  d'où ce modèle est à changement de point.

### Méthode MCMC

Le problème consiste à estimer les paramètres  $\theta$ ,  $\lambda$  ainsi le paramètre  $k$  par l'approche MCMC.

**La vraisemblance :**

$$\begin{aligned} L(\theta, \lambda, k|y) &= \prod_{i=1}^k \frac{\theta^{y_i}}{(y_i)!} e^{-\theta} \times \prod_{i=k+1}^{112} \frac{\lambda^{y_i}}{(y_i)!} e^{-\lambda} \\ &\propto \theta^{\sum_{i=1}^k y_i} e^{-k\theta} \lambda^{\sum_{i=k+1}^{112} y_i} e^{-(112-k)\lambda} \\ &= \left(\frac{\theta}{\lambda}\right)^K e^{k(\lambda-\theta)} \lambda^{\sum_{i=1}^{112} y_i} e^{-112\lambda}, \end{aligned}$$

où  $K = \sum_{i=1}^k y_i$ .

Lois a priori :

$$\theta \sim \mathcal{G}(a_1, b_1);$$

$$\lambda \sim \mathcal{G}(a_2, b_2);$$

$$k \in \{1 \dots 112\}.$$

Lois conditionnelles :

$$\theta|y, \lambda, k \sim \mathcal{G}(a_1 + \sum_{i=1}^k y_i, b_1 + k);$$

$$\lambda|y, \theta, k \sim \mathcal{G}(a_2 + \sum_{i=k+1}^{112} y_i, b_2 + 112 - k);$$

$$p(k|y, \theta, \lambda) = \frac{Q(y, k, \theta, \lambda)}{\sum_{j=1}^{112} Q(y, j, \theta, \lambda)};$$

où

$$Q(y, k, \theta, \lambda) = e^{k(\lambda-\theta)} \left(\frac{\theta}{\lambda}\right)^K.$$

Puisque on connaît les lois conditionnelles il est facile d'appliquer l'échantillonneur de Gibbs.

### Algorithme

- Initialiser les paramètres  $a_1, a_2, b_1, b_2$  ;
- Soit  $Y$  l'échantillon des données citées auparavant qui suit la loi de poisson à changement de point ;
- À l'étape  $t$ , Tirer  $\theta$  à partir de la densité  $\mathcal{G}(a_1 + \sum_{i=1}^k y_i, b_1 + k)$  ;
- Tirer  $\lambda$  à partir de la densité  $\mathcal{G}(a_2 + \sum_{i=k+1}^{112} y_i, b_2 + 112 - k)$  ;
- Tirer  $k$  on utilisant  $\theta$  et  $\lambda$  générés à cette étape par la densité conditionnelle de  $k$  ;
- Incrimenter  $t$  à  $t + 1$ . (Voir L'annexe)

### Résultats

Les estimateurs des paramètres  $\theta, \lambda$  et  $k$  obtenus avec un nombre de période de chauffe est 100 sur 1100 itérations sont :

$\theta$	$\lambda$	$k$
3.001287	0.97414	40.0675

Les représentations graphiques des résultats sont données dans les figures (5.3), (5.4), (5.6), (5.7), (5.9) et (5.10) :

**Paramètre  $\theta$**

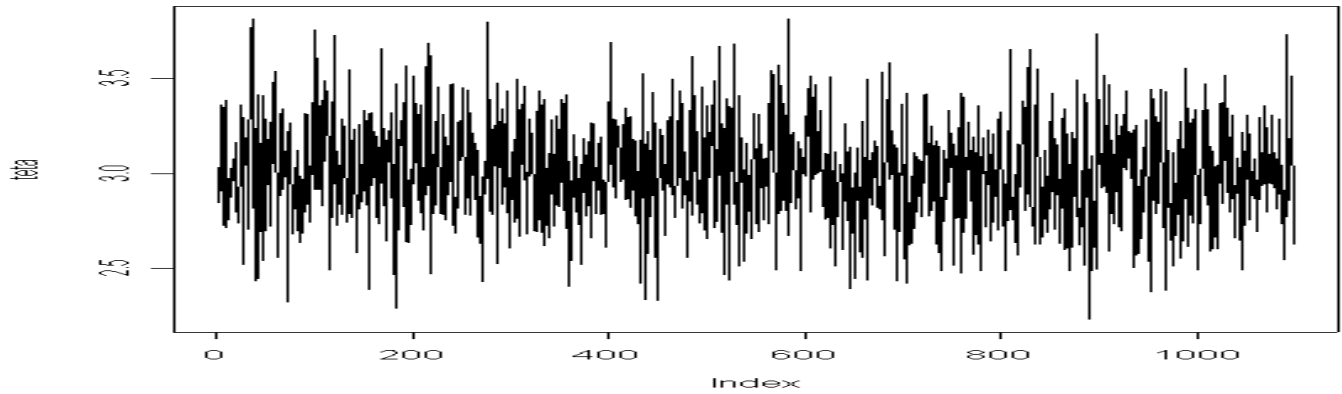


FIGURE 4.57 – Evolution de la chaîne du paramètre ( $\theta$ )

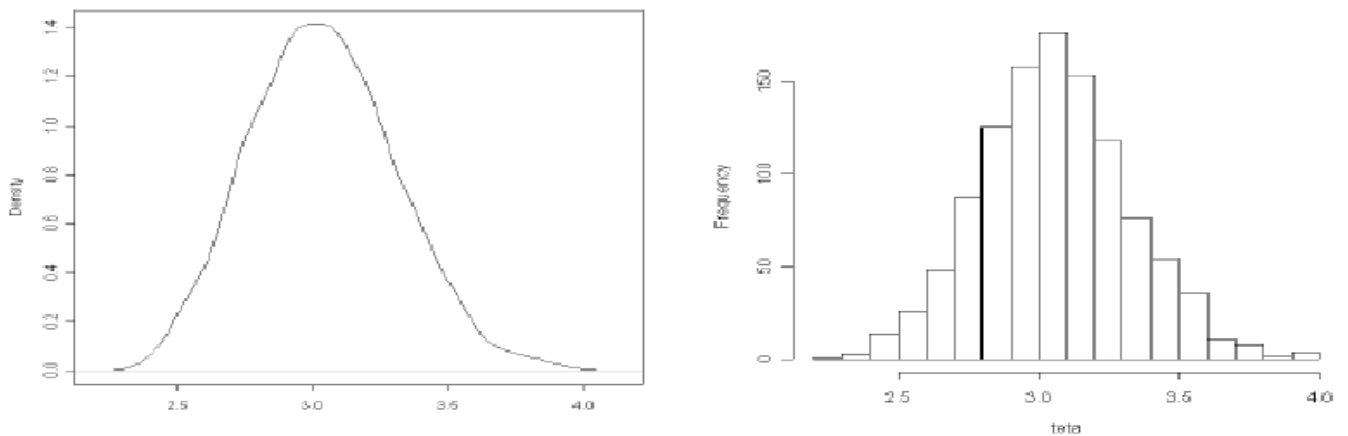


FIGURE 4.58 – Densité et histogramme du paramètre ( $\theta$ )

**Etude de la convergence**

Pour vérifier la convergence du paramètre  $\theta$ , nous avons calculé le facteur d'échelle en utilisant  $M = 3$  et un nombre d'itérations  $N = (100, 500, 3000)$ .

Les résultats sont donnés dans le tableau suivant :

Nombre de Chaînes	Nombre d'itérations	Facteur d'échelle
3	100	1.0109
3	500	1.0102
3	3000	1.0021

TABLE 4.11 – Facteur d'échelle du paramètre ( $\theta$ )

La représentation graphique de la convergence de l'estimateur du paramètre  $\theta$  est donnée comme suit :

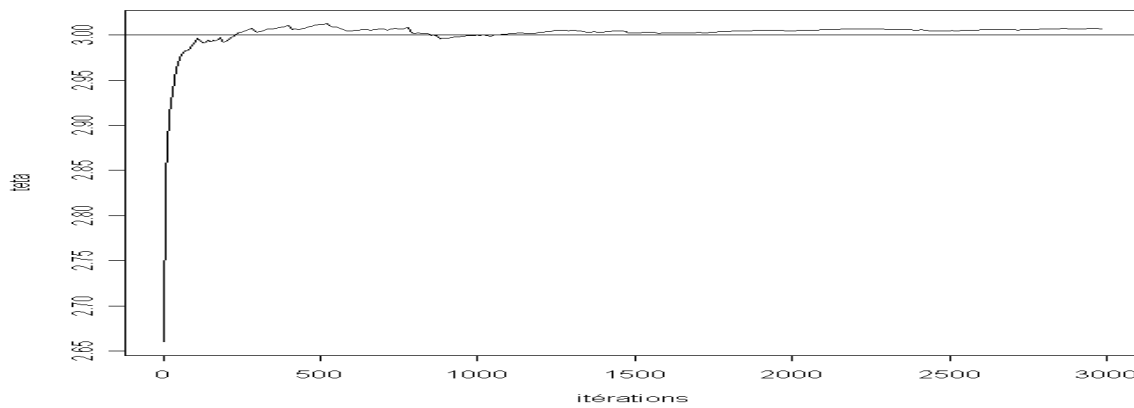


FIGURE 4.59 – Convergence de l'estimateur du paramètre ( $\theta$ )



Paramètre  $\lambda$

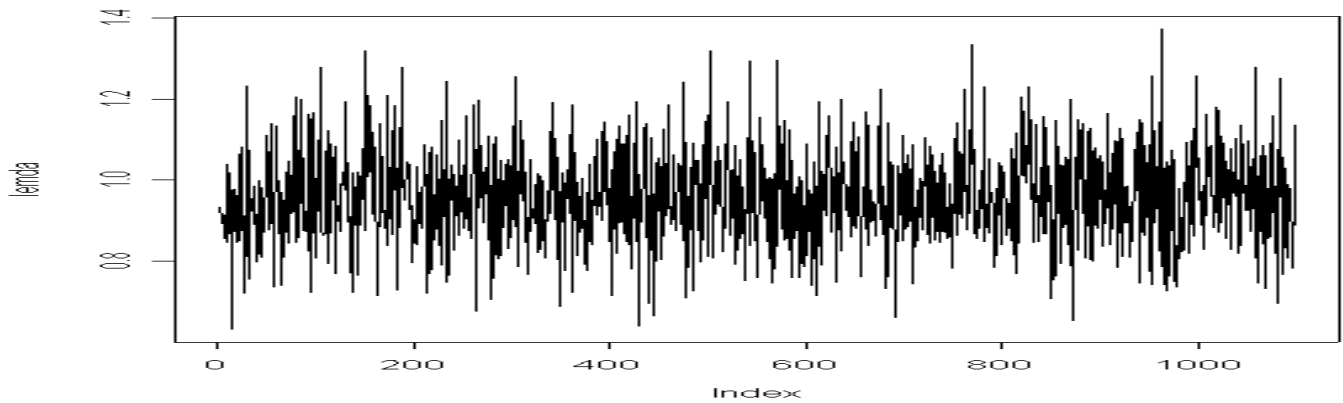


FIGURE 4.60 – Evolution de la chaîne du paramètre ( $\lambda$ )

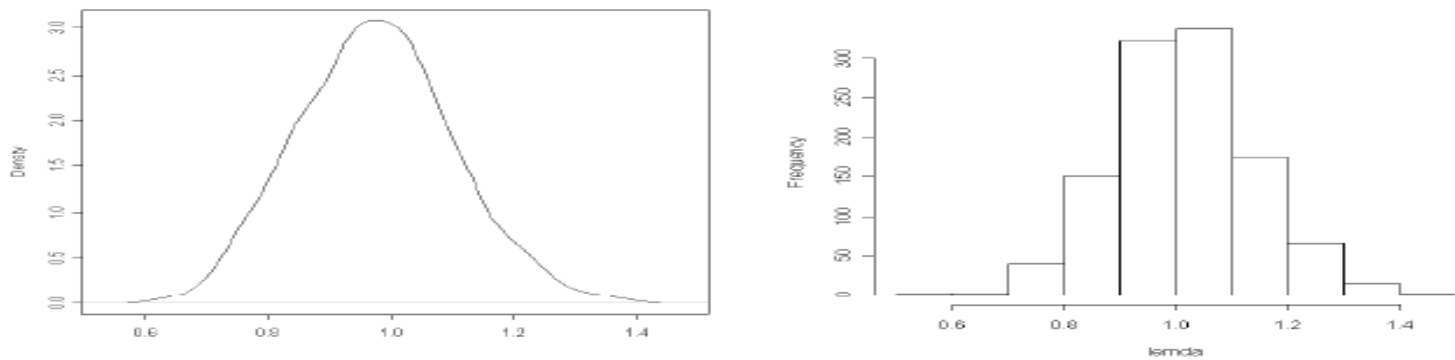


FIGURE 4.61 – Densité et histogramme du paramètre ( $\lambda$ )

Les résultats de l'étude de convergence pour le paramètre  $\lambda$  sont donnés dans le tableau suivant :

Nombre de Chaînes	Nombre d'itérations	Facteur d'échelle
3	100	1.145
3	500	1.105
3	3000	1.006

TABLE 4.12 – Facteur d'échelle du paramètre ( $\lambda$ )

La représentation graphique de la convergence de l'estimateur du paramètre  $\lambda$  est donnée comme

suit :

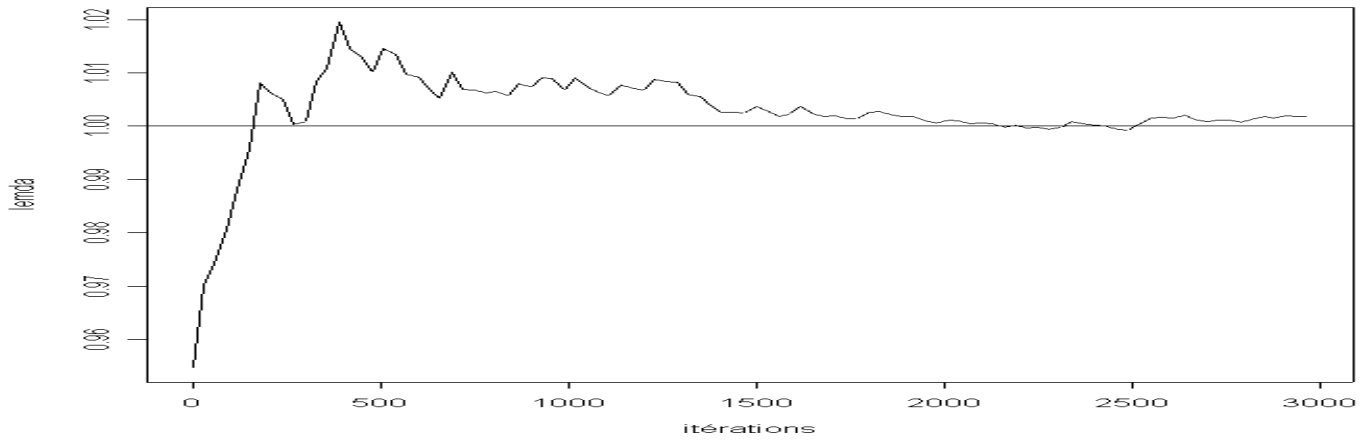


FIGURE 4.62 – Convergence de l’estimateur du paramètre ( $\lambda$ )

Paramètre  $k$

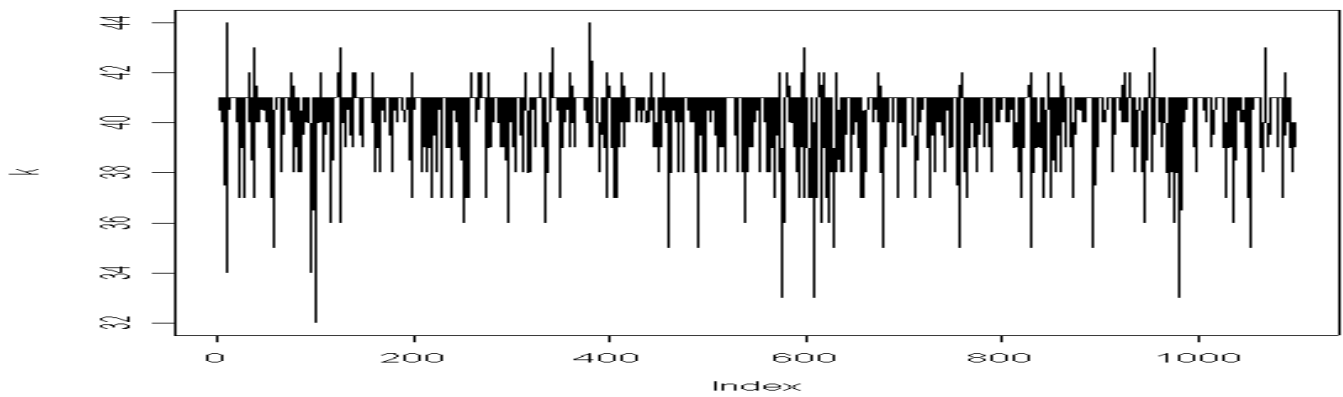


FIGURE 4.63 – Evolution de la chaîne du paramètre ( $k$ )

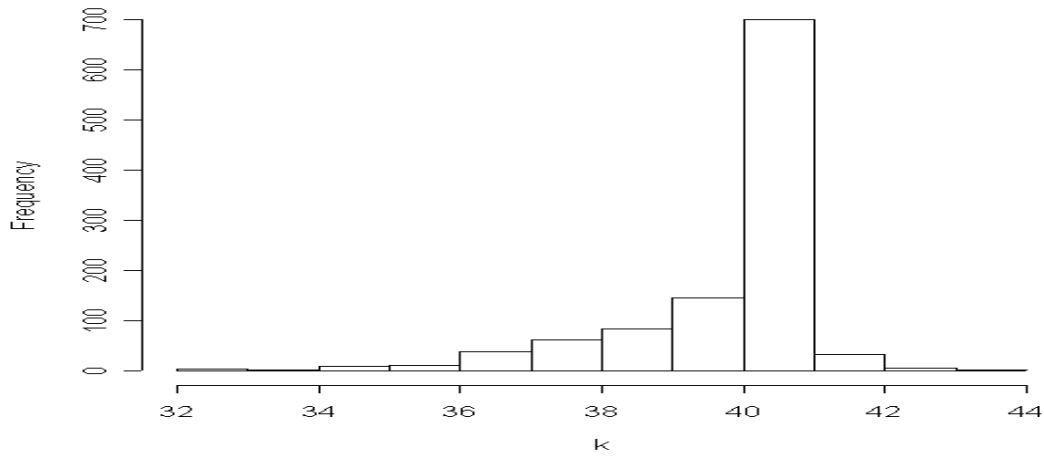


FIGURE 4.64 – Histogramme du paramètre ( $k$ )

Les résultats de l'étude de convergence pour le paramètre  $k$  sont donnés dans le tableau suivant :

Nombre de Chaînes	Nombre d'itérations	Facteur d'échelle
3	100	1.1582
3	500	1.1078
3	3000	1.0679

TABLE 4.13 – Facteur d'échelle du paramètre ( $k$ )

La représentation graphique de la convergence de l'estimateur du paramètre  $k$  est donnée dans la figure suivante :

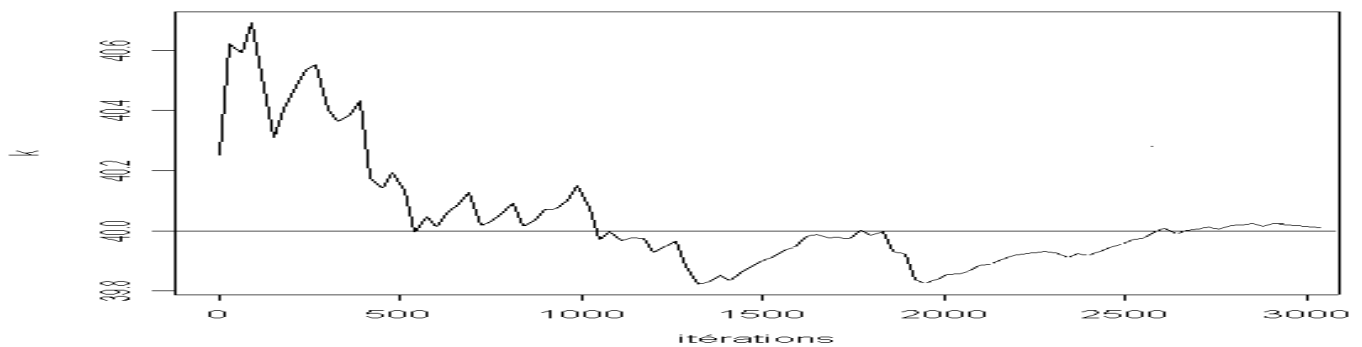


FIGURE 4.65 – Convergence de l'estimateur du paramètre ( $k$ )

## Méthode PMC

Le problème consiste à estimer les paramètres  $\theta$ ,  $\lambda$  ainsi le paramètre  $k$  par l'approche PMC.

### Loi a posteriori :

$$\pi(\theta, \lambda, k|y) \propto L(\theta, \lambda, k|y) * \theta^{a_1-1} e^{-b_1\theta} * \lambda^{a_2-1} e^{-b_2\lambda} * \frac{1}{112}.$$

L'algorithme utilisé est celui de PMC général qui consiste à :

- Initialiser les paramètres  $a_1, a_2, b_1, b_2$  ;
- Soit  $Y$  l'échantillon des données citées auparavant ;
- Les lois de propositions sont des lois *Normale* tronquées sur  $[1, \infty[$  pour les paramètres ( $\theta$  et  $\lambda$ ) et pour  $k$  à estimer par la méthode (*MH*) ;
- À l'étape  $t$  : générer un vecteur de taille  $M$  pour le paramètre  $\theta$  ainsi pour  $\lambda$  selon les lois de propositions, et utiliser ces paramètres pour générer  $k$ , ensuite calculer le poids de chaque particule ainsi les poids normalisés et rééchantillonner selon ces derniers. (Voire l'annexe )

## Résultats

Les estimateurs des paramètres  $\theta, \lambda$  et  $k$  obtenus, avec le nombre d'itérations  $T = 100$  sont :

$\theta$	$\lambda$	$k$
3.057205	1.067044	40.09722

Les représentations graphiques des résultats sont données dans les figures (5.12), (5.13), (5.15), (5.16), (5.18) et (5.19) :

### Paramètre $\theta$

La représentation graphique de la convergence de l'estimateur du paramètre  $\theta$  est donnée dans la figure suivante :

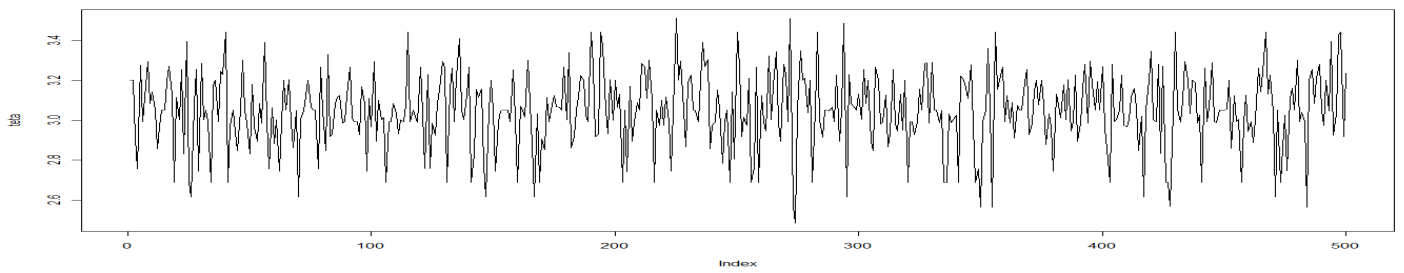


FIGURE 4.66 – Evolution de la chaîne du paramètre ( $\theta$ )

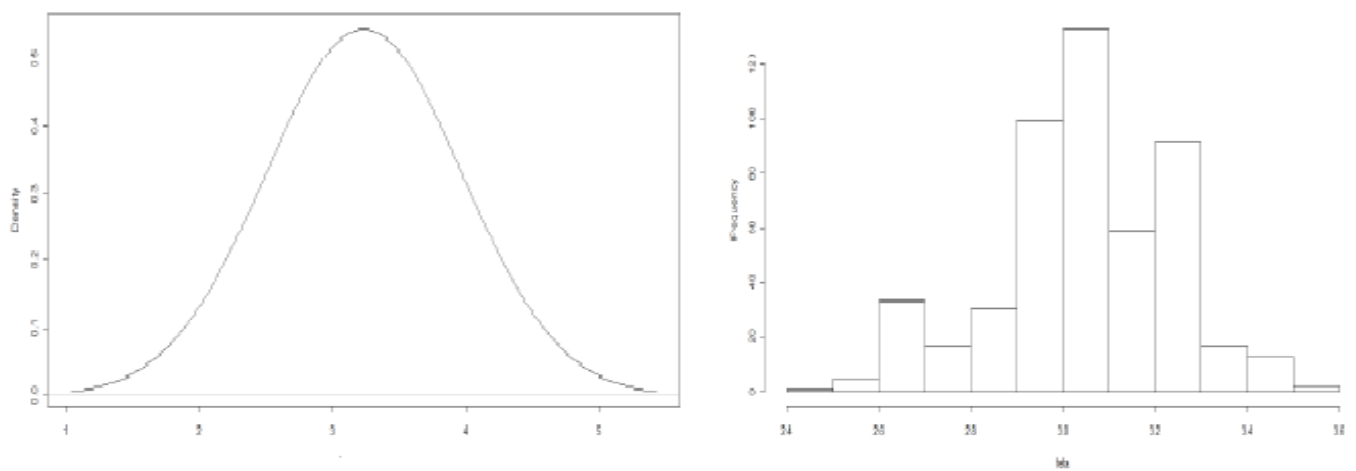


FIGURE 4.67 – Densité et histogramme de la chaîne du paramètre ( $\theta$ )

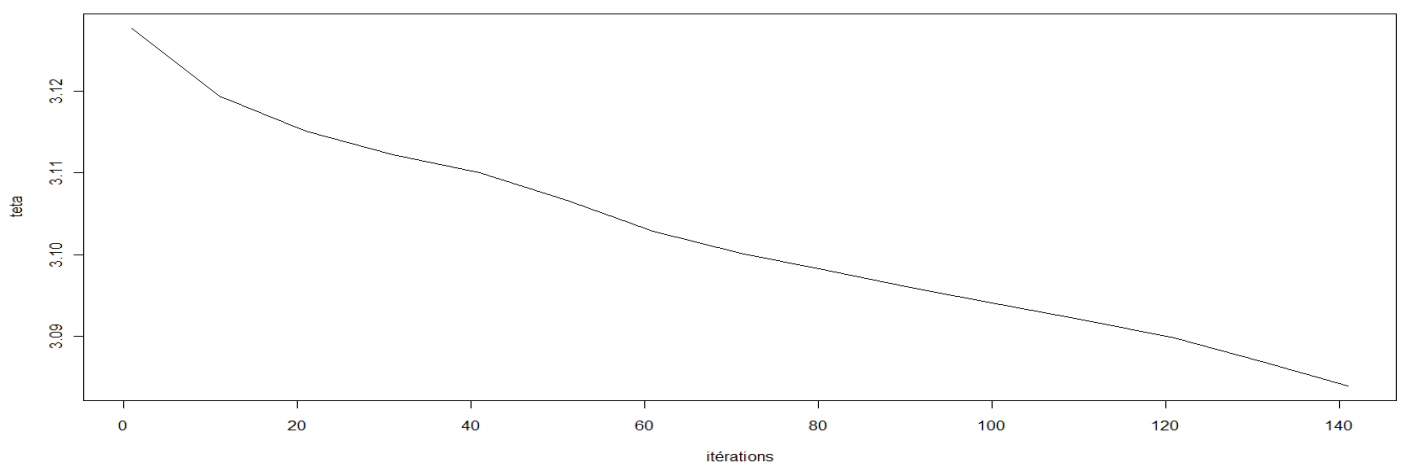


FIGURE 4.68 – Convergence de l'estimateur du paramètre ( $\theta$ )

Paramètre  $\lambda$

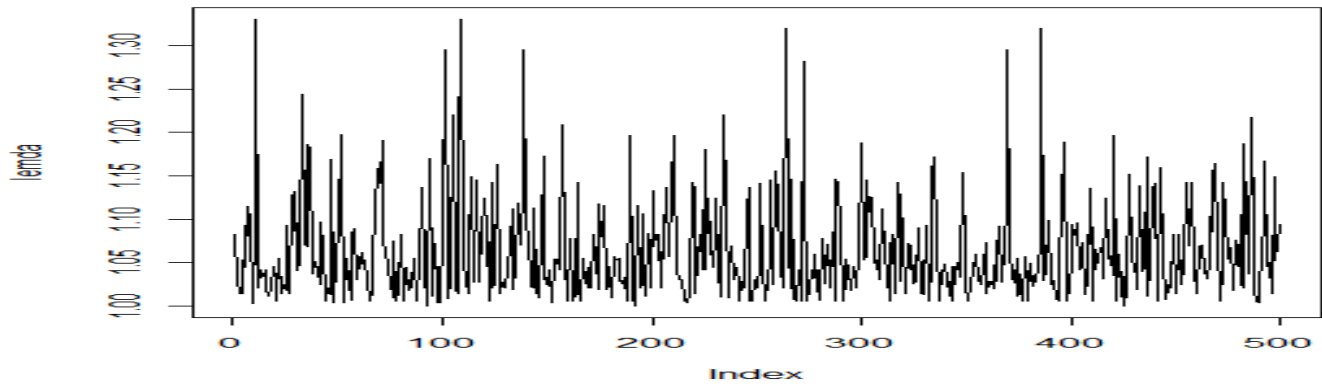


FIGURE 4.69 – Evolution de la chaîne du paramètre ( $\lambda$ )

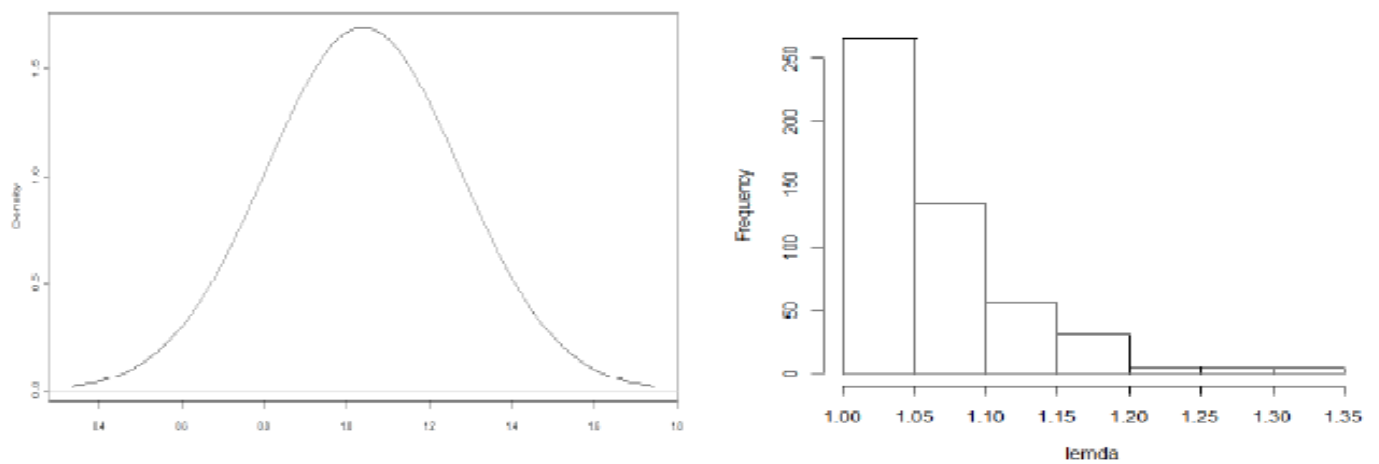


FIGURE 4.70 – Densité et histogramme de la chaîne du paramètre ( $\lambda$ )

La représentation graphique de la convergence de l'estimateur du paramètre  $\lambda$  est donnée dans la figure suivante :

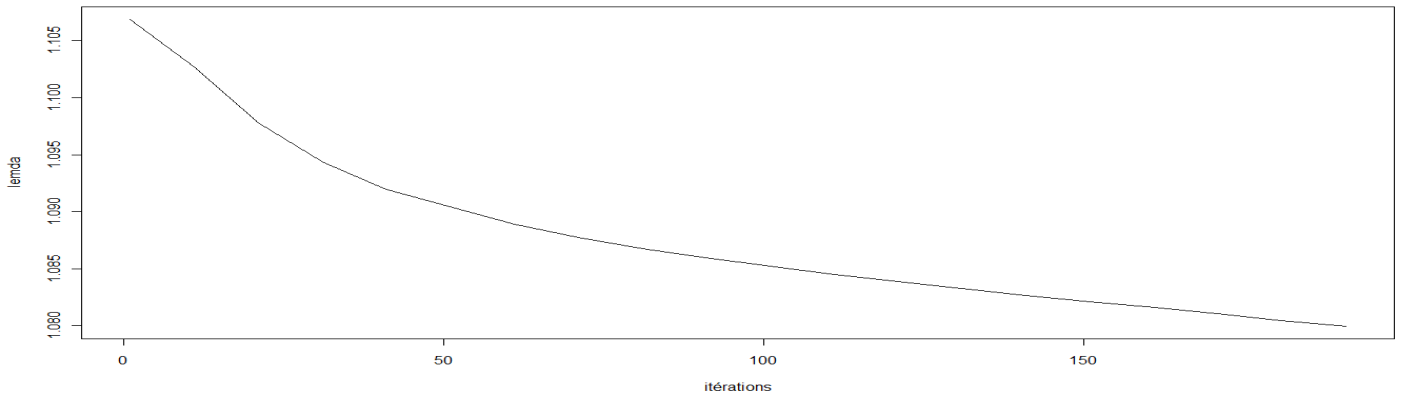


FIGURE 4.71 – Convergence de l'estimateur du paramètre ( $\lambda$ )

**Paramètre  $k$**

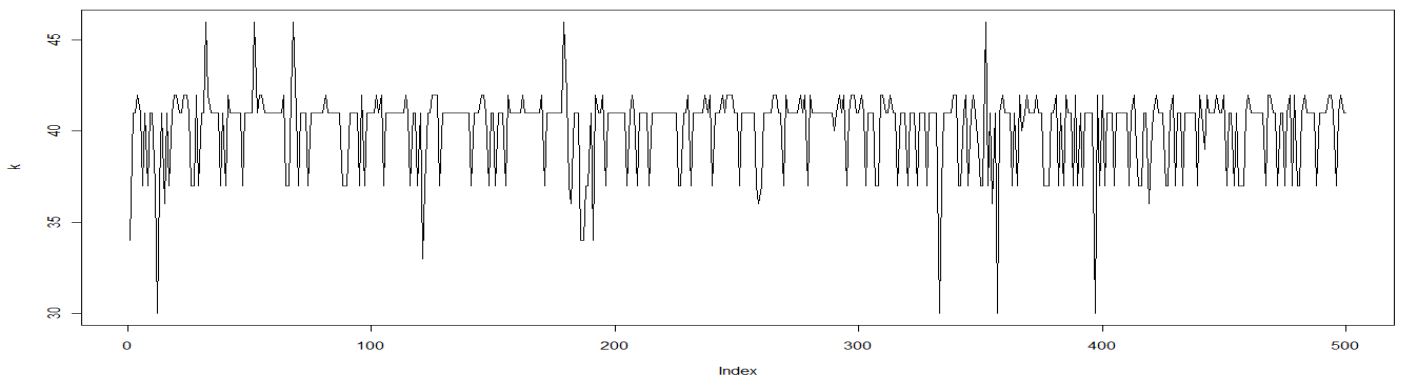


FIGURE 4.72 – Evolution de la chaîne du paramètre ( $k$ )

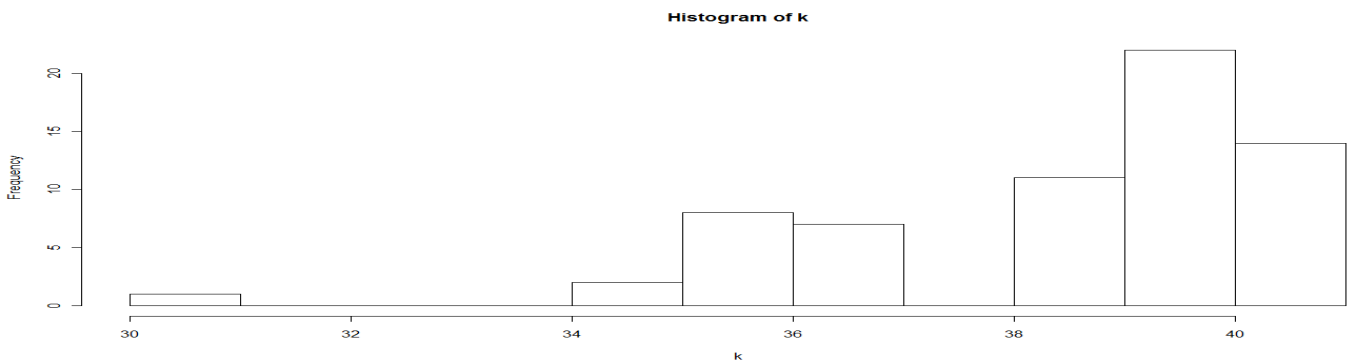


FIGURE 4.73 – Histogramme de la chaîne du paramètre ( $k$ )

La représentation graphique de la convergence de l'estimateur du paramètre  $k$  est donnée dans la figure suivante :

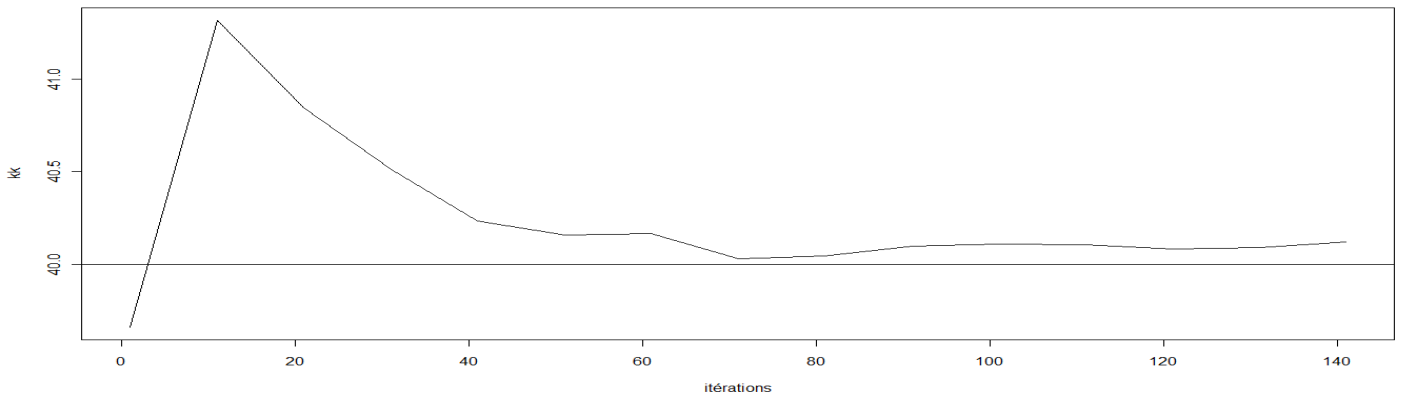


FIGURE 4.74 – Convergence de l'estimateur du paramètre ( $k$ )

## 4.12 Discussion

Les résultats des estimateurs  $\theta$  et  $\lambda$  obtenus montrent qu'il y a une diminution de nombre de catastrophes, du fait que le paramètre  $\theta$  est supérieur à celui de  $\lambda$ . Il y a une grande évidence que le changement de point est en ( $t = 40$ ), qui représente le mode du paramètre  $k$  estimé. Ce qui signifie que ces catastrophes commencent vraiment à diminuer à partir de l'année 1890.

Les résultats montrent que pour la méthode MCMC, les estimateurs sont obtenus à partir d'un nombre d'itérations égal à 1100. La convergence de ces paramètres est atteinte à partir d'un nombre d'itérations ( $> 300$ ) pour le paramètre  $\theta$ , ( $> 700$ ) pour le paramètre  $\lambda$  et ( $> 500$ ) pour le paramètre  $k$ . La méthode PMC, fournit d'excellentes approximations pour un nombre d'itérations ( $T = 100$ ) pour les trois paramètres. La convergence de ces derniers est obtenue à partir d'un nombre d'itérations ( $> 90$ ) pour le paramètre  $\theta$  et  $\lambda$  et ( $> 70$ ) pour le paramètre  $k$ .

Les performances de PMC (9.80s) sont meilleures que celles de MCMC (13.11s) au sens de temps d'exécution.



# Conclusion Générale

L'objectif de ce travail est de présenter et comparer les principes des algorithmes de la méthodes Monte Carlo par Chaîne de Markov(MCMC) et Population de Monte Carlo (PMC) qui sont des outils de simulation stochastiques très puissants permettant de résoudre une grande partie des problèmes d'inférences statistiques.

Dans la première partie de ce mémoire, nous avons cité les différents algorithmes de l'approche MCMC, ainsi qu'une méthode de diagnostique de convergence. En pratique, on ne peut simuler qu'un nombre fini d'itérations de MCMC, et le choix des conditions initiales peut influencer la qualité de convergence. Il est alors nécessaire de ne pas tenir compte des premières itérations de l'algorithme, mais il est difficile de déterminer la durée de cette période de "chauffe" ("burn-in period"). Il n'existe pas de critère rigoureux de convergence permettant de savoir si la chaîne est proche de son régime asymptotique, il existe en revanche des techniques empiriques. De plus, la vitesse de convergence de ces algorithmes est très faible. L'algorithme de Gibbs est le plus rapide, mais il ne peut pas être utilisé dans toutes les situations.

Dans la deuxième partie, nous avons présenté les algorithmes de la méthode PMC, qui est une généralisation de la technique de l'échantillonnage préférentiel tout en supprimant la difficulté fondamentale d'évaluation de la convergence.

Nous avons montré dans le chapitre 4, que le schéma PMC est meilleur que MCMC. Pour ce faire, nous avons utilisé différents modèles comme (Gaussien, Bernoulli, Log Normal,...), pour lesquels la méthodes MCMC ne fournit pas des approximations satisfaisantes sauf pour un nombre d'itérations grand. Par contre pour PMC nous avons constaté, qu'après un nombre de simulations relativement réduit, elle fournit d'excellentes approximations des paramètres estimés.

On a également effectué dans le dernier chapitre une analyse sur des données concernant le nombre de catastrophes causés par le charbonnage aux niveaux des mines de Grande-Bretagne durant une période

de 112 années. Après une étude effectuée sur ces données, le modèle proposé par Carlin et al.(1992) est celui de Poisson a changement de point. Le but d'utilisation des méthodes MCMC est PMC est d'estimer les paramètres du modèle proposé. On a appliqué ces deux méthodes sur cet exemple réel où on confirme la rapidité de l'approche PMC a celle de MCMC en temps d'exécution et en nombre de simulation pour l'estimation des paramètres.

# Annexe

## A. Code source R

Les programmes informatiques ont été implimentés sous *R*. Nous présentons les programmes mis en place pour la comparaison des méthodes MCMC et PMC Ainsi les programmes de la partie application.

### A.1. Comparaison des deux méthodes

#### A.1.1. Modèle gaussien

```
=====  
Estimation par MCMC  
=====
```

```
library(MCMCpack)  
gibbs=function(T)  
{  
x=rnorm(500,0,1)  
m0=0.01  
sig0=1  
a0=1  
b0=0.5  
tau=0  
m=0
```

```

tau[1]=1
n=length(x)
for(i in 1:T)
{
sig=sqrt((tau[i]*sig0)/(tau[i]+n*sig0))
mea=(tau[i]/(tau[i]+n*sig0))*m0+(n*sig0/(tau[i]+n*sig0))*mean(x)
m[i]=rnorm(1,mea,sig)
a=a0+(n/2)
b=b0+0.5*sum((x-mu[i])^2)
tau[i+1]=rinvgamma(1,a,b)
}
print(mean(m))
print(mean(tau))
}

```

```

=====
                Estimation par PMC
=====

```

```

library(MCMCpack)
library(msm)
pmc=function(T){
M=500
m0=0.01
sig0=0.2
a0=1
b0=0.2
x=rnorm(500,0,1)
n=length(x)
m=matrix(0,nrow=T+1, ncol=M)
sigma=matrix(0,nrow=T+1, ncol=M)
r=matrix(0,nrow=T+1, ncol=M)
w=matrix(0,nrow=T+1, ncol=M)
vrais=matrix(0,nrow=T+1, ncol=M)

```

```
histm=0
histsigm=0
vrai=0
t<-1
m[1,]<-rnorm(M,1,0.3) #le vecteur initial de m
sigm[1,]<-rinvgamma(M,1,0.2) #vecteur initial de sigma
while (t<=T){
m[t+1,]=rnorm(M,m[t,],0.03)
sigm[t+1,]=rtnorm(M,sigm[t,],0.03,0,Inf)
for(i in 1:m){
vrai[i]=prod(dnorm(x,m[t+1,i],sigm[t+1,i]))
}
r[t+1,]=(vrai*dnorm(m[t+1,],m0,sig0)*dinvgamma(sigm[t+1,],a0,b0))/(dnorm(m[t+1,],teta[t,],0.03)
*dtnorm(sigm[t+1,],sigm[t,],0.03,0,Inf))
r[t+1,]=(r[t+1,]/sum(r[t+1,]))
m[t+1,]=sample(m[t+1,], M, replace = TRUE, prob =r[t+1,])
sigm[t+1,]=sample(sigm[t+1,], M, replace = TRUE, prob =r[t+1,])

histm=histm+sum(r[t+1,]*m[t+1,])
histsigm=histsigm+sum(r[t+1,]*sigm[t+1,])
t=t+1
}
histfinalm=(1/T)*histm
histfinalsigm=(1/T)*histsigm
print(histfinalm)
print(histfinalsigm)
}
```

## A.1.2. Modèle Binomial

```
=====
Estimation par MCMC
=====
```

```
f=function(t){
z=t^(39)*(1-t)^(61)
return(z)
}
binomial=function(T){
p=array(dim = c(T))
p[1]=1
for (i in 1:T){
yi=runif(1,0,(p[i]+1)) # loi de proposition
d=f(yi)*dunif(p[i],0,(yi+1))
e=f(p[i])*dunif(yi,0,(p[i]+1))
k=d/e
prob=min(1,k)
u=runif(1)
if (u < prob ){
p[i+1] = yi
}else{
p[i+1] = p[i]
}
}
return (mean(p))
}
```

```
=====
Estimation par PMC
=====
```

```
f=function(t){
```

```

z=t^(39)*(1-t)^(61)
return(z)
}
binomialpmc=function(T){
M=500
p=matrix(0,nrow=T+1, ncol=M)
r=matrix(0,nrow=T+1, ncol=M)
histp=0
t<-1
p[1,<-runif(M) # vecteur initial
while (t<=T){
p[t+1,]=runif(M,p[t,],1) #loi de proposition
r[t+1,]=f(p[t+1,])/dunif(p[t+1,],p[t,],1)
r[t+1,]=(r[t+1,]/sum(r[t+1,]))
p[t+1,]=sample(p[t+1,], M, replace = TRUE, prob =r[t+1,])
histp=histp+sum(r[t+1,]*p[t+1,])
t=t+1
}
histfinalp=(1/T)*histp
return(histfinalp)
}

```

### A.1.3. Modèle Log normale

```

=====
Estimation par MCMC
=====

```

```

library(MCMCpack)
gibbs=function(T)
{
x=rnorm(500,0,1)
mulog=0

```

```

taulog=0
m0=0.1
sig0=1
a0=1
b0=0.5
tau=0
m=0
tau[1]=1
n=length(x)

for(i in 1:size)
{
sig=sqrt((tau[i]*sig0)/(tau[i]+n*sig0))
mea=(tau[i]/(tau[i]+n*sig0))*m0+(n*sig0/(tau[i]+n*sig0))*mean(x)
m[i]=rnorm(1,mea,sig)
a=a0+(n/2)
b=b0+0.5*sum((x-mu[i])^2)
tau[i+1]=rinvgamma(1,a,b)
mlog[i]=exp(0.5*(mu[i]+tau[i+1]))
taulog[i]=(exp(tau[i+1])-1)*exp(2*mu[i]+tau[i+1])
}
print(mean(mlog))
print(mean((taulog))
}

=====

          Estimation par PMC

=====

library(MCMCpack)
library(msm)
pmc1=function(T){
M=500
m0=0.01

```



---

```

sig0=0.5
a0=1
b0=0.2
x=rnorm(500,0,1)
n=length(x)
m=matrix(0,nrow=T+1, ncol=M)
sigm=matrix(0,nrow=T+1, ncol=M)
r=matrix(0,nrow=T+1, ncol=M)
w=matrix(0,nrow=T+1, ncol=M)
vrai=matrix(0,nrow=T+1, ncol=M)
histm=0
histsigm=0
vrai=0
t<-1
m[1,]<-rnorm(M,1,0.3)
sigm[1,]<-rinvgamma(M,1,0.2)
while (t<=T){
m[t+1,]=rnorm(M,teta[t,],0.03)
sigm[t+1,]=rtnorm(M,sigm[t,],0.03,0,Inf)
for(i in 1:m){
vrai[i]=prod(dnorm(x,m[t+1,i],sigm[t+1,i]))
}
r[t+1,]=(vrai*dnorm(teta[t+1,],m0,sig0)*dinvgamma(sigm[t+1,],a0,b0))/(dnorm(teta[t+1,],teta[t,],
*dtnorm(sigm[t+1,],sigm[t,],0.03,0,Inf))
r[t+1,]=(r[t+1,]/sum(r[t+1,]))
m[t+1,]=sample(m[t+1,], M, replace = TRUE, prob =r[t+1,])
sigm[t+1,]=sample(sigm[t+1,], M, replace = TRUE, prob =r[t+1,])
histm=histm+sum(r[t+1,]*m[t+1,])
histsigm=histsigm+sum(r[t+1,]*sigm[t+1,])
t=t+1
}
histfinalm=(1/T)*histm

```

---

```
histfinalsigm=(1/T)*histsigm
mlog=exp(0.5*(histfinalm+histfinalsigm))
sigmlog=(exp(histfinalsigm)-1)*exp(2*histfinalm+histfinalsigm)
}
```

#### A.1.4. Modèle Poisson hiérarchique

```
=====
                Estimation par MCMC
=====
```

```
library(MCMCpack)
pgibbs=function(T){
a1=1
a2=1
b1=1
b2=1
n=112
y1=0
y2=0
teta=0
lemda=0
b=0
bb=0
k=100
for(i in 1:n)
{
if (i < k)
{
y=rpois(1,1)
y1=y1+y
}
}
}
}
```

```
y=rpois(1,3)
y2=y2+y
}
}
y1
y2
for(i in 1:T)
{
a=a1+y1
b=b1+k
aa=a2+y2
bb=b2+(n-k)
teta[i]=rgamma(1,a,b)
lemda[i]=rgamma(1,aa,bb)
}
print(mean(teta))
print(mean(lemda))
}
```

```
=====
                Estimation par PMC
=====
```

```
library(MCMCpack)
library(msm)
poisson=function(T){
a1=1
a2=1
b1=1
b2=1
n=112
y1=0
y2=0
M=500
```

```
teta=0
lemda=0
b=0
bb=0
k=100
for(j in 1:n)
{
if (j <= k)
{
y=rpois(1,1)
y1=y1+y

}else {
y=rpois(1,3)
y2=y2+y
}
}
sy=y1+y2
teta=matrix(0,nrow=T+1, ncol=M)
lemda=matrix(0,nrow=T+1, ncol=M)
r=matrix(0,nrow=T+1, ncol=M)
histeta=0
histlemda=0
post=0
teta[1,]<-runif(M)
lemda[1,]<-runif(M)
t<-1
while (t<=T){
teta[t+1,]=rtnorm(M,teta[t,],0.5,1,Inf)
lemda[t+1,]=rtnorm(M,lemda[t,],0.1,1,Inf)
for(i in 1:M){
post[i]=(teta[t+1,i]/lemda[t+1,i])^(y1)*exp(k*(lemda[t+1,i]-teta[t+1,i]))
```

```

*(lemda[t+1,i])^(sy)*exp(-n*(lemda[t+1,i]))
}
r[t+1,]=(post*teta[t+1,]^(a1-1)*exp(-b1*teta[t+1,])*lemda[t+1,]^(a2-1)*exp(-b2*lemda[t+1,])
*(1/112))/(dtnorm(teta[t+1,],teta[t,],0.5,1,Inf)*dtnorm(lemda[t+1,],lemda[t,],1.1,0,Inf)*k)
r[t+1,]=(r[t+1,]/sum(r[t+1,]))
teta[t+1,]=sample(teta[t+1,], M, replace = TRUE, prob =r[t+1,])
lemda[t+1,]=sample(lemda[t+1,], M, replace = TRUE, prob =r[t+1,])
histeta=histeta+sum(r[t+1,]*teta[t+1,])
histlemda=histlemda+sum(r[t+1,]*lemda[t+1,])
t=t+1
}
histfinalteta=(1/T)*histeta
histfinalemda=(1/T)*histlemda
print(histfinalteta)
print(histfinalemda)
}

```

### A.1.5. Modèle de Bernoulli

```

=====
                Estimation par MCMC
=====
bernoulli=function(T)
{
x=rbinom(500,1,0.5)
a=1
b=2
p=0
n=length(x)
s=a+sum(x)
ss=b+n-sum(x)
for(i in 1:T)

```

```

{
p[i]=rbeta(1,s,ss)
}
print(mean(p))
}

=====

          Estimation par PMC

=====

bernollipmc=function(T)
{
x=rbinom(500,1,0.5)
M=1000
a=1
b=2
n=length(x)
p=matrix(0,nrow=T+1, ncol=M)
r=matrix(0,nrow=T+1, ncol=M)
w=matrix(0,nrow=T+1, ncol=M)
vrais=matrix(0,nrow=T+1, ncol=M)
histp=0
vrai=0
t<-1
p[1,]<-runif(M)  ##le vecteur initial
while (t<=T){
#print(t)
p[t+1,]=rnorm(M,p[t,],0.00003)
for(i in 1:M){
vrais[i]=prod(dbinom(x,1,p[t+1,i]))
}
r[t+1,]=(vrais*dbeta(p[t+1,],a,b))/(dnorm(p[t+1,],p[t,],0.00003))
r[t+1,]=(r[t+1,]/sum(r[t+1,]))
p[t+1,]=sample(p[t+1,], M, replace = TRUE, prob =r[t+1,])

```

```
histp=histp+sum(r[t+1,]*p[t+1,])
t=t+1
}
hitimateur=(1/T)*histp
print(hitimateur)
}
```

### A.1.6. Modèle Exponentiel à données manquantes

```
=====
```

Estimation par MCMC

```
=====
```

```
gibbsmanq=function(T)
{
n=20
a=0.1
b=0.1
teta=0
z=0
teta[1]=1
s=0
c=0.4
r=12
y=rexp(n,0.8519)
for(i in 1:(n-r)){
s=s+y[i]
}
A=a+n
B=b+s+r*c
for(i in 1:T)
{
z[i]=rgamma(1,r,teta[i])
```

```
teta[i+1]=rgamma(1,A,(B+z[i]))
```

```
}
```

```
print(mean(teta))
```

```
}
```

```
=====
```

Estimation par PMC

```
=====
```

```
PMCmanquante=function(T)
```

```
{
```

```
n=20
```

```
m=200
```

```
a=0.1
```

```
b=0.1
```

```
rr=12
```

```
s=0
```

```
c=0.4
```

```
y=rexp(n,0.8519)
```

```
for(i in 1:(n-rr)){
```

```
s=s+y[i]
```

```
}
```

```
A=a+n-rr
```

```
B=b+s+rr*c
```

```
AA=a+n
```

```
tita=matrix(0,nrow=T+1, ncol=m)
```

```
z=matrix(0,nrow=T+1, ncol=m)
```

```
r=matrix(0,nrow=T+1, ncol=m)
```

```
w=matrix(0,nrow=T+1, ncol=m)
```

```
nt=matrix(0,nrow=T+1, ncol=m)
```

```
dt=matrix(0,nrow=T+1, ncol=m)
```

```
vrai=matrix(0,nrow=T+1, ncol=m)
```

```
histeta=matrix(0,nrow=T+1, ncol=1)
```

```
hteta=0
```



```

t<-1
tita[1,]=runif(m,0.6,1)
while (t<=T){
for(i in 1:m){
z[t,]=rgamma(1,rr,tita[t,i])
tita[t+1,]=rgamma(m,n+a,s+(rr*c)+b+z[t,i])
}
for(i in 1:m){
for(j in 1:m){
nt[t+1,j]=dgamma(tita[t+1,j],A,B)*dgamma(tita[t+1,i],a,b)/dgamma(z[t,j],rr,tita[t,j])
dt[t+1,j]=dgamma(z[t,j],rr,tita[t,i])*dgamma(tita[t+1,i],n+a,s+(rr*c)+b+z[t,j])/dgamma(z[t,j],rr,tita[t,j])
}
som1=sum(nt[t+1,])
som2=sum(dt[t+1,])
ntt=(1/m)*som1
dtt=(1/m)*som2
r[t+1,]=ntt/dtt
r[t+1,]=(r[t+1,]/sum(r[t+1,]))
}
tita[t+1,]=sample(tita[t+1,], m, replace = TRUE, prob = r[t+1,])
hteta=hteta+sum(r[t+1,]*tita[t+1,])
t=t+1
}
histeta=(1/T)*hteta
print(histeta)
hist(tita)
}

```

## A.2. Partie application

=====

Estimation par MCMC

```

=====
library(MCMCpack)
#####
Q=function(q,t,l,gk){
Z=exp(q*(1-t))*(t/l)^(gk)
return(Z)
}
#####
pgibbs=function(T){
a1=1
a2=1
b1=1
b2=1
k=0
yi=0
n=112
teta=0
lemda=0
b=0
bb=0
k[1]=60
s=0
z=0
t<-1
yi=rpois(T,70)
y=c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,
5,5,3,4,2,5,2,2,3,4,2,1,3,2,2,1,1,1,1,3,0,0,1,0,1
,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,
2,3,3,1,1,2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1)
while (t<=T){
for(i in 1:k[t]){
s= s+y[i]

```

```

}
for(i in (k[t]+1):n){
z=z+y[i]
}
a=a1+s
b[t]=b1+k[t]
aa=a2+z
bb[t]=b2+(n-k[t])
teta[t]=rgamma(1,a,b)
lemda[t]=rgamma(1,aa,bb)
d=Q(yi[t],teta[t],lemda[t],s)
e=Q(k[t],teta[t],lemda[t],s)
l=d/e
prob=min(1,l)
u=runif(1)
if (u < prob ){
  k[t+1] = yi[t]
}else{
k[t+1]=k[t]
}
t=t+1
s=0
z=0
}
print(mean(teta))
print(mean(lemda))
print(mean(k))
print(names(which.max(table(k))))
}

```

```

=====
Estimation par PMC
=====

```

Le programme qui estime le paramètre  $k$  par la methode *MCMC* (*MH*) est le suivant :

```
#####
Q=function(q,t,l,gk){
Z=exp(q*(1-t))*(t/l)^(gk)
return(Z)
}
#####
mcmcgibs=function(A,B,y1,ka,yi)
{
d=Q(yi,A,B,y1)
e=Q(ka,A,B,y1)
l=d/e
prob=min(1,l)
u=runif(1)
if (u < prob ){
ka = yi
}else{
ka=ka
}
return(ka)
}
```

Le programme principal est le suivant :

```
library(MCMCpack)
library(msm)
mp=function(T){
a1=1
a2=1
b1=1
b2=1
n=112
y1=0
```

---

```

y2=0
k=0
m=500
teta=0
lemda=0
b=0
bb=0
teta=matrix(0,nrow=T+1, ncol=m)
lemda=matrix(0,nrow=T+1, ncol=m)
k=matrix(0,nrow=T+1, ncol=m)
r=matrix(0,nrow=T+1, ncol=m)
hteta=0
hlemda=0
hk=0
vrais=0
t<-1
y=c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,
5,5,3,4,2,5,2,2,3,4,2,1,3,2,2,1,1,1,1,3,0,0,1,0,1
,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,
2,3,3,1,1,2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1)
teta[1,]<-rpois(m,5)  ##le vecteur initial de teta
lemda[1,]<-rpois(m,3)  ##vecteur initial de lemda
k[1,]=rpois(m,40)  ##vecteur initial de k
yi=rpois(m,70)
while (t<=T){
teta[t+1,]=rtnorm(m,teta[t,],0.1,1,Inf)
lemda[t+1,]=rtnorm(m,lemda[t,],0.1,1,Inf)
for(i in 1:m){
s=0
ss=0
for(j in 1:k[t,i])
{

```

```

s=s+y[j]
}
s
for(j in (k[t,i]+1):112)
{
ss=ss+y[j]
}
ss
sss=s+ss
k[t+1,i]=mcmcgibs(teta[t+1,i],lemda[t+1,i],s,k[t,i],yi[i])
vrais[i]=(teta[t+1,i]/lemda[t+1,i])^(s)*exp(k[t+1,i]*(lemda[t+1,i]-teta[t+1,i]))*(lemda[t+1,i])^(1-s)
}
r[t+1,]=(vrais*teta[t+1,]^(a1-1)*exp(-b1*teta[t+1,])*lemda[t+1,]^(a2-1)*exp(-b2*lemda[t+1,]))/(
(dtnorm(teta[t+1,],teta[t,],0.1,1,Inf)*dtnorm(lemda[t+1,],lemda[t,],0.1,1,Inf))*k[t+1,])
r[t+1,]=(r[t+1,]/sum(r[t+1,]))
teta[t+1,]=sample(teta[t+1,], m, replace = TRUE, prob =r[t+1,])
lemda[t+1,]=sample(lemda[t+1,], m, replace = TRUE, prob =r[t+1,])
k[t+1,]=sample(k[t+1,], m, replace = TRUE, prob =r[t+1,])
hteta=hteta+sum(r[t+1,]*teta[t+1,])
hlemda=hlemda+sum(r[t+1,]*lemda[t+1,])
hk=hk+sum(r[t+1,]*k[t+1,])
t=t+1
}
ssteta=(1/T)*hteta##l'estimateur de teta
sslemda=(1/T)*hlemda
ssk=(1/T)*hk
print(ssteta)
print(sslemda)
print(ssk)
print(names(which.max(table(k))))
}

```

# Bibliographie

- [1] P. Baldi, L. Mazliak, and P. Priouret. *Martingales et chaînes de Markov*. Décembre 1998.
- [2] M. Benaïm and N. El Karoui. *Promenade aléatoire : Chaînes de Markov et simulation ; martingales et stratégies*. 91128 Palaiseau Cedex, École polytechnique edition, Novembre 2007.
- [3] J. Bernier and E. Parent. *Le raisonnement bayésien, modélisation et inférence*. Springer edition, 2007.
- [4] J. Bégin. Analyse mcmc de certains modèles de diffusion avec application au marché européen du carbone. Technical report, Université de Montréal, Montréal, Canada, Août 2010.
- [5] N. Bousquet. Analyse bayésienne de la durée de vie de composants industriels. Décembre 2006.
- [6] O. Cappé, A. Guillin, J.M. Marin, and C.P. Robert. Population monte carlo. *J. Comput. Graph. Statist.*, 13, 2004.
- [7] B.P. Carlin, A.E. Gelfand, and Smith. Hierarchical bayesian analysis of changepoint problems. *J. Roy. Stat. Soc.*, C(41) :389–405, 1992.
- [8] G. Celeux, J.M. Marin, and C.P. Robert. Iterated importance sampling in missing data problems. 50 :3386–3404, Août 2005.
- [9] J.S. Dagpunar. *Simulation and Monte Carlo : With applications in finance and MCMC*. John Wiley and Sons Ltd, England, 2007.
- [10] D. Flipo. Chaînes de markov. *Université des Sciences et Technologies de Lille, U. F. R. de Mathématiques Pures et Appliquées*.
- [11] A. Gelman, W.R. Gilks, and G.O. Roberts. Efficient metropolis jumping rules. *Bayesian Statistics*, 5, 1996.
- [12] A. Gelman and D.B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4) :457–472, Novembre 1992.

- 
- [13] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. 6 :721–741, 1984.
- [14] A. Guillin, J.M. Marin, and C.P. Robert. *Estimation bayésienne approximative par échantillonnage préférentiel*. rapport de recherche, Université Paris IX Dauphine, 2005.
- [15] W.K. Hastings. *Monte Carlo Sampling methods using Markov chains and their applications*, volume 57. Biometrika, April 1970. pages 97-109.
- [16] J. Jacod. Chaîne de markov, processus de poisson et applications. Dea de probabilités et applications, 2003-2004.
- [17] R.G. Jarrett. A note on the intervals between coal-mining disasters. *Biometrika*, 66 :191–193, 1979.
- [18] R.L. Karandikar. On the markov chain monte carlo (mcmc) method. Technical report, Indian Statistical Institute, New Delhi, April 2006.
- [19] V. Mazet. *Développement de méthodes de traitement de signaux spectroscopiques : estimation de la ligne de base et du spectre de raies*. Spécialité automatique, traitement du signal et génie informatique, Université Henri Poincaré, Nancy 1, Décembre 2005.
- [20] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computings machines. *Chemical Physics*, 21(6) :1087–1092, June 1953.
- [21] C. Robert. *Méthodes de Monte Carlo par Chaînes de Markov*. 49 Rue Héricart, 75015 Paris, economica edition, 1996.
- [22] A. Ruegg. *Processus stochastique*. Presses polytechniques romandes edition.
- [23] B. Shababa. Bayesian analysis. *STATS 225*, Janvier 2011.
- [24] L. Tierney. Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 1994.
- [25] J.Y. Tourneret. Méthodes de monte-carlo par chaînes de markov. *Institut de recherche en informatique de Toulouse (IRIT)*, Novembre 2007.
- [26] M. Vieira. *Estimation Bayésienne par des méthodes MCMC : Application à la surveillance des moteurs asynchrones*. Thèse de doctorat en science, Université de Nice-Sophia-Antipolis, 41 Bd Napoléon III 06041 Nice Cedex, Janvier 1999.
- [27] Q. Yao. Tests for change-points with epidemic alternatives. *Biometrika*, 80 :179,191, 1993.



## *Résumé*

Dans ce travail, nous comparons sur des données simulées et sur des données réelles, les méthodes Monte Carlo par Chaîne de Markov (MCMC) et Population Monte Carlo (PMC).

La méthode MCMC introduite par Metropolis et al. en 1953 puis généralisée par Hastings en 1970. Elle se subdivise en deux : l'algorithme de Metropolis Hastings et l'échantillonneur de Gibbs qui est un cas particulier de ce dernier.

Population Monte Carlo (PMC) est le très récent et puissant algorithme introduit initialement par Cappé et al. en (2004) puis amélioré par Douc et al. en (2005) et très récemment Celeux et al. en (2006) qui ont également mis à profit cette méthode pour étudier des modèles à données manquantes.

Les résultats obtenus montrent que la méthode PMC est meilleure que MCMC en temps d'exécution et en nombre de simulations .

**Mots clés :** MCMC, PMC, Chaîne de Markov, Monte Carlo, estimation bayésienne.

## *Abstract*

In this work, we compare on simulated data and real data, the methods Monte Carlo Markov Chain(MCMC) and Population Monte Carlo (PMC).

The method MCMC introduced by Metropolis and al in 1953 then generalized by Hastings in 1970. They are subdivided in two principal types : algorithm of Metropolis Hastings and the sampler of Gibbs which is a particular case of this last.

Population Monte Carlo (PMC) is the very recent one and a powerful algorithm introduced initially by Cappé and al in (2004) then improved by Douc and al in (2005). Recently in (2006) Celeux and al also made this method profitable to study models with a missing data.

The obtained results show that the method PMC is better than the method MCMC in speed.

**Key words :** MCMC, PMC, Markov Chains, Monte Carlo, Bayesian estimation.