

Master Recherche Biostatistiques

Université de Montpellier II

ANALYSE BAYÉSIENNE DE MODÈLES D'ÉVOLUTION DE RESSOURCES NATURELLES

par

Marie Denis

sous la direction de

Fabien Campillo ¹

et

Vivien Rossi ²

Soutenu le **18 juin 2007** devant le jury composé de :

C. ABRAHAM	Professeur, Montpellier SupAgro
A. BERLINET	Professeur, Université Montpellier II
J.P. DAURÈS	Professeur, Université Montpellier I
G. DUCHARME	Professeur, Université Montpellier II

¹Projet ASPI (IRISA/INRIA)

²UR Dynamique des forêts naturelles (CIRAD)

Remerciements

Le stage s'est réalisé au sein de l'unité de recherche «dynamique des forêts naturelles» au CIRAD de Baillarguet où le travail est décontracté et studieux.

Je tiens tout d'abord à remercier **Fabien Campillo** et **Vivien Rossi** pour leurs disponibilités, leurs conseils et surtout leurs patiences. J'ai acquis à leurs côtés un savoir faire qui servira pour le futur.

Mes remerciements se dirigent également à mes collègues de bureau : **Pierrette** et **Frédéric Mortier** pour leurs conseils, **Guillaume Cornu** pour son aide précieuse en informatique, sans oublier **Émilie** pour sa bonne humeur.

Table des matières

Remerciements	1
Introduction	1
1 Modèles markoviens	3
1.1 Chaînes de Markov	3
1.2 Modèles de Markov cachés	4
2 Filtre optimal	6
2.1 Filtre bayésien optimal et modélisation du problème	6
2.2 État de l'art des méthodes classiques	7
2.2.1 Le filtre de Kalman	8
2.2.2 Filtre de kalman étendu	13
3 Filtrage particulaire	17
3.1 Filtre de Monte Carlo et Bootstrap	18
3.2 Filtres SIS et SIR	25
3.3 Améliorations	27
3.3.1 Filtre particulaire auxiliaire (ASIR)	27
3.3.2 Méthodes de Monte Carlo par chaînes de Markov et filtres particuliers	30
3.3.3 Amélioration de l'échantillonnage séquentiel de Monte Carlo	33
4 Applications	38
4.1 Application à la pêcheerie	38
4.1.1 Données et méthodes	38
4.1.2 Résultats	40

TABLE DES MATIÈRES **3**

4.2	Application à la foresterie	44
4.2.1	Introduction	44
4.2.2	Données et méthodes	44
4.2.3	Mise en œuvre et résultats	49
Conclusion		61
Annexes		63
A Algorithmes		63
A.1	Ellipse de confiance	63
A.2	Redistribution des résidus	63
B Données de pêche		65
Bibliographie		66

Introduction

L'inférence bayésienne pour les modèles à espaces d'état couplée aux méthodes de Monte Carlo par Chaînes de Markov (MCMC) est, depuis quelques années, de plus en plus utilisée dans le domaine de l'environnement. Le cadre bayésien se prête bien à ce domaine d'application. En effet, les données sont peu nombreuses et très souvent bruitées. Il est donc nécessaire de faire appel aux compétences des spécialistes du domaine afin, d'une part, de proposer des systèmes dynamiques modélisant l'évolution des phénomènes considérés et d'autre part, d'apporter de l'information sur les paramètres inconnus du modèle sous forme de loi a priori.

L'utilisation des méthodes MCMC nécessitent une attention particulière sur deux points : le diagnostic de convergence et l'accélération de la convergence. Ces deux aspects causent une certaine incertitude sur les résultats obtenus. Aussi les utilisateurs des méthodes MCMC, en particulier dans l'environnement, sont intéressés par d'autres techniques qui permettraient une validation de leurs résultats. Les techniques de filtrage particulière, développées dans le milieu des années 1990, en font partie. Ces méthodes ont été à l'origine développées pour des problèmes où de nombreuses observations sont à traiter en temps réel, comme par exemple le domaine de la robotique ou bien encore de la poursuite de cible. Les techniques particulières ont le même objectif que les MCMC, à savoir l'approximation de la loi a posteriori et sont basées sur le même principe de simulation de Monte Carlo. Cependant, elles sont séquentielles, c'est à dire elles traitent les données en ligne et sont ainsi plus rapides. Ces méthodes ne sont pas conçues pour le contexte environnemental où la fréquence d'acquisition des données est grande. La rapidité du traitement n'est plus un enjeu essentiel.

L'objet de ce mémoire est d'adapter ces méthodes particulières à des modèles d'évolution de ressources renouvelables et ainsi de mettre à disposition, des écologues, une nouvelle méthode pour évaluer leurs résultats.

Le mémoire s'articule de la façon suivante.

Dans le premier chapitre nous définissons le cadre de cet exposé, le modèle de Markov caché, au travers de rappels sur les chaînes de Markov.

Dans le deuxième chapitre nous présentons le problème du filtrage, en exposant le filtre optimal bayésien et les différentes techniques d'approximation dites "classiques". Puis dans le troisième chapitre nous abordons le cœur de ce travail : le problème du filtrage particulière. Nous discutons des algorithmes de base et des améliorations possibles. Le contexte est différent du contexte habituel du filtrage particulière, car il n'y a plus de contraintes de temps réel. On peut s'autoriser des variantes plus lentes et ainsi utiliser des algorithmes qui n'ont pas été mis en pratique jusqu'à présent.

Le dernier chapitre est consacré à une application dans le domaine de l'halieutique, il s'agit de reconstituer l'évolution de la biomasse du poulpe de Mauritanie entre 1971 et 2004. Cette application est traitée dans le cadre d'une collaboration avec Étienne Rivot (Laboratoire d'Écologie Halieutique de l'AGROCAMPUS Rennes) et de l'ARC MICR. Les données ont été transmises par Étienne Rivot et récoltées par la Délégation à la Surveillance Pêche et au Contrôle en Mer (DSPCM). Dans le cadre de cette collaboration l'évolution de la biomasse du poulpe a été estimée par une méthode MCMC dans le groupe de travail d'Étienne Rivot. Parallèlement, nous avons utilisé le filtre Bootstrap et l'ASIR pour effectuer cette même estimation. Nous présentons nos résultats ainsi que la comparaison avec les résultats obtenus par la méthode MCMC.

Chapitre 1

Modèles markoviens

1.1 Chaînes de Markov

Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé et $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ un espace d'état, on appelle *processus stochastique* $X = (X_k)_{k \geq 0}$ à valeurs dans \mathbb{R}^n une suite de variables aléatoires à valeurs dans \mathbb{R}^n . On utilisera la notation $X_{0:k} = (X_0, \dots, X_k)$. La loi de ce processus est la famille de probabilité $\mathbb{P}(X_{0:k} \in dx_{0:k})$ pour tout $k \geq 0$.

Définition 1.1.1 (Chaîne de Markov) *Un processus $X = (X_k)_{k \geq 0}$ à valeurs dans \mathbb{R}^n est appelé chaîne de Markov (ou processus de Markov) s'il satisfait à la propriété de Markov suivante :*

$$\mathbb{P}(X_{k+1} \in dx | X_{0:k}) = \mathbb{P}(X_{k+1} \in dx | X_k), \quad \forall k \geq 0.$$

Ce qui signifie donc que le futur conditionné par le passé est le futur conditionné par le présent.

On obtient grâce à la définition 1.1.1 la factorisation de la loi du processus :

$$\begin{aligned} \mathbb{P}(X_{0:k} \in dx_{0:k}) &= \mathbb{P}(X_k \in dx_k | X_{k-1} = x_{k-1}) \mathbb{P}(X_{k-1} \in dx_{k-1} | X_{k-2} = x_{k-2}) \cdots \\ &\quad \cdots \mathbb{P}(X_1 \in dx_1 | X_0 = x_0) \mathbb{P}(X_0 \in dx_0) \\ &= \mathbb{P}(X_0 \in dx_0) \prod_{i=1}^k \mathbb{P}(X_i \in dx_i | X_{i-1} = x_{i-1}). \end{aligned}$$

La loi d'un processus de Markov est complètement déterminée par sa loi initiale :

$$\mu_0(dx) \stackrel{\text{déf}}{=} \mathbb{P}(X_0 \in dx)$$

et par son noyau de transition :

$$Q_k(x, dx') \stackrel{\text{déf}}{=} \mathbb{P}(X_k \in dx' | X_{k-1} = x).$$

La loi de transition détermine l'évolution de la loi de la variable X_k .

Hypothèse 1.1.2 (Homogénéité en temps) *Dans la suite de notre exposé nous supposons que notre processus est homogène en temps, c'est à dire que le noyau de transition ne dépend pas de k , ce qui s'écrit :*

$$Q(x, dx') = \mathbb{P}(X_k \in dx' | X_{k-1} = x), \quad \forall k \geq 0.$$

Exemple 1.1.3 *La solution de l'équation suivante est un processus de Markov :*

$$X_{k+1} = f_k(X_k, W_k),$$

où $X_0, (W_k)_{k \geq 0}$ sont indépendants et $(W_k)_{k \geq 0}$ est une suite de variables aléatoires indépendantes de moyenne nulle. Dans le cas particulier où le bruit est additif et $W_k \sim \mathcal{N}(0, \sigma_w^2 I)$, on a :

$$Q(x, dx') = \mathcal{N}(f_{k-1}(x), \sigma_w^2 I).$$

1.2 Modèles de Markov cachés

Soit un processus de Markov $(X_k)_{k \geq 0}$ de loi initiale μ_0 et de noyau de transition Q . Supposons que nous n'ayons pas accès à ce processus, on l'observe donc via un second processus $(Y_k)_{k \geq 1}$ à valeurs dans \mathbb{R}^d , dit processus d'observation. Les variables Y_k sont supposées indépendantes conditionnellement au processus d'état et leur distribution dépend uniquement de l'état courant, cette hypothèse est appelée *hypothèse de canal sans mémoire*, ce qui s'écrit :

$$\mathbb{P}(Y_{1:t} \in dy_{1:t} | X_{0:t} = x_{0:t}) = \prod_{i=1}^t \mathbb{P}(Y_i \in dy_i | X_i = x_i).$$

Sous les hypothèses de Markov et de canal sans mémoire la loi du processus (X_k, Y_k) est complètement déterminée par :

– Sa loi initiale :

$$\mu_0(dx) \stackrel{\text{déf}}{=} \mathbb{P}(X_0 \in dx),$$

– Sa loi de transition :

$$Q(x, dx') \stackrel{\text{déf}}{=} \mathbb{P}(X_k \in dx' | X_{k-1} = x),$$

– Ses noyaux d'émission :

$$\Psi_k(x, dy) \stackrel{\text{déf}}{=} \mathbb{P}(Y_k \in dy | X_k = x), \quad \forall k \geq 0.$$

On supposera que le noyau d'émission est, tout comme la loi de transition, homogène en temps. On a donc que :

$$\Psi(x, dy) = \mathbb{P}(Y_k \in dy | X_k = x), \quad \forall k \geq 0.$$

Ainsi la loi du modèle s'écrit :

$$\begin{aligned} & \mathbb{P}(X_{0:t} \in dx_{0:t}, Y_{1:t} \in dy_{1:t}) \\ &= \mathbb{P}(Y_{1:t} \in dy_{1:t} | X_{0:t} = x_{0:t}) \mathbb{P}(X_{0:t} \in dx_{0:t}) \\ &= \mathbb{P}(X_0 \in dx_0) \prod_{i=1}^t \mathbb{P}(Y_i \in dy_i | X_i = x_i) \mathbb{P}(X_i \in dx_i | X_{i-1} = x_{i-1}) \\ &= \mu_0(dx_0) \prod_{i=1}^t \Psi(x_i, dy_i) Q(x_{i-1}, dx_i). \end{aligned}$$

Hypothèse 1.2.1 (Hypothèse de domination) Une hypothèse de domination est nécessaire pour définir la vraisemblance du modèle. On suppose donc l'absolue continuité du noyau d'émission par rapport à une mesure ν , i.e. $\Psi(x, dy) \ll \nu(dy)$ pour tout x . On note ψ la densité correspondante :

$$\Psi(x, dy) = \psi(x, y) \nu(dy).$$

Exemple 1.2.2 (Système non linéaire à espace d'état) Le système suivant est un exemple de modèle de Markov caché à espace d'état continu.

$$\begin{aligned} X_{k+1} &= f(X_k) + g(X_k) W_k, \\ Y_k &= h(X_k) + V_k, \end{aligned}$$

où X_0, W_k, V_k sont indépendants et

$$W_k \sim q_k^W(w) dw, \quad V_k \sim q_k^V(v) dv.$$

Dans la cas particulier où $V_k \sim \mathcal{N}(0, Q_k^V)$, on a que :

$$\psi(x, y) \propto \exp\left(-\frac{1}{2} (y - h(x))^* (Q_k^V)^{-1} (y - h(x))\right).$$

Chapitre 2

Filtre optimal

Dans ce second chapitre nous présentons la solution théorique du problème de filtrage : le filtre optimal bayésien. Dans le cas simple où le problème est linéaire et gaussien, le filtre de Kalman permet d'obtenir une solution explicite, de nombreux livres détaillent cette méthode, voir par exemple [7], [11]. Dans le cas non linéaire, le filtre optimal bayésien n'est pas accessible, nous présentons les différentes méthodes dites "classiques" permettant de résoudre ce problème.

2.1 Filtre bayésien optimal et modélisation du problème

On définit le système dynamique composé des deux processus $(X_k)_{k \geq 0}$ à valeurs dans \mathbb{R}^n et $(Y_k)_{k \geq 1}$ à valeurs dans \mathbb{R}^d . On suppose les propriétés de Markov et de canal sans mémoire vérifiées.

Le problème du filtrage consiste à calculer à chaque pas de temps la loi conditionnelle π_k , appelée *filtre*, de l'état X_k sachant une réalisation des observations $Y_{1:k} = y_{1:k}$ jusqu'à l'instant courant k , i.e. :

$$\pi_k(dx) = \mathbb{P}(X_k \in dx | Y_{1:k} = y_{1:k}). \quad (2.1)$$

La suite de probabilités conditionnelles $(\pi_k)_{k \geq 0}$, avec la convention $\pi_0 = \mu_0$ constitue le *filtre optimal bayésien*. Pour des contraintes de temps réel, on souhaite calculer ce filtre de façon récursive, i.e. de telle façon que le calcul de π_k ne soit fonction que de la loi conditionnelle précédente π_{k-1} et de

l'observation Y_k . L'évolution du filtre optimal dans le temps est relativement facile à décrire, elle se décompose en une étape de prédiction et une étape de correction faisant intervenir la nouvelle observation.

Proposition 2.1.1 (Filtre optimal) *Le filtre se calcule de façon récursive en deux étapes. Le filtre prédit π_{k-} (dx) = $\mathbb{P}(X_k \in dx | Y_{1:k-1})$ à l'étape k se déduit du filtre π_{k-1} selon l'équation de prédiction :*

$$\pi_{k-}(dx) = \int_{\mathbb{R}^n} \pi_{k-1}(dx') Q_k(x', dx) = (\pi_{k-1} Q_k)(dx). \quad (2.2)$$

Le filtre π_k se déduit à l'étape k du filtre prédit π_{k-} selon l'équation de correction :

$$\pi_k(dx) = \frac{\psi_k(x, y_k) \pi_{k-}(dx)}{\int_{\mathbb{R}^n} \psi_k(x', y_k) \pi_{k-}(dx')} = (\psi_k \cdot \pi_{k-})(dx). \quad (2.3)$$

L'évolution du filtre peut se résumer par le schéma suivant :

$$\begin{array}{ccccc} & & & Y_k & \\ & & & \downarrow & \\ \pi_{k-1} & \xrightarrow{\text{prédiction}} & \pi_{k-} = (Q_k \pi_{k-1}) & \xrightarrow{\text{correction}} & \pi_k = (\psi_k \cdot \pi_{k-}) \end{array}$$

TAB. 2.1 – Décomposition d'une itération du filtre

Ce filtre est une équation récursive dans un espace infini-dimensionnel et ne peut donc pas être utilisé en pratique. Dans certains cas, comme le cas linéaire-gaussien, ce filtre admet une formulation explicite. Dans le cas général il est nécessaire de faire appel à des méthodes d'approximation.

2.2 État de l'art des méthodes classiques

Nous présentons dans cette section un historique des algorithmes classiques de filtrage linéaire puis non linéaire. Nous verrons les méthodes analytiques, à savoir le filtre de Kalman et ses dérivées : le filtre de Kalman étendu et le filtre de Kalman inodore (Unscented Kalman filter, dénoté *UKF* en anglais) ; ce dernier filtre, dont le but est de se passer des calculs de gradient qui apparaissent dans le filtre de Kalman étendu en faisant appel à des formules de quadrature, est proposé dans [2].

2.2.1 Le filtre de Kalman

On considère un modèle linéaire et gaussien :

$$X_{k+1} = F_k X_k + G_k W_k, \quad \forall k \geq 0, \quad (2.4)$$

$$Y_k = H_k X_k + V_k, \quad \forall k \geq 1, \quad (2.5)$$

où $X_0 \sim \mathcal{N}(\bar{X}_0, Q_0)$, $W_k \sim \mathcal{N}(0, Q_k^W)$, $V_k \sim \mathcal{N}(0, Q_k^V)$ et X_0, W_k, V_k sont indépendants. On suppose que $Q_k^V \geq 0$. La sortie de ce système étant gaussienne, la loi conditionnelle π_k est donc également gaussienne. On pose $\pi_k = \mathcal{N}(\hat{X}_k, R_k)$ et $\pi_{k-} = \mathcal{N}(\hat{X}_{k-}, R_{k-})$ avec :

$$\begin{aligned} \hat{X}_k &= \mathbb{E}[X_k | Y_{1:k}], & R_k &= \mathbb{E}[(X_k - \hat{X}_k)(X_k - \hat{X}_k)^*], \\ \hat{X}_{k-} &= \mathbb{E}[X_k | Y_{1:k-1}], & R_{k-} &= \mathbb{E}[(X_k - \hat{X}_{k-})(X_k - \hat{X}_{k-})^*]. \end{aligned}$$

L'idée est toujours la même : on veut déterminer la loi π_k en deux étapes. La prédiction : on calcule π_{k-} à l'aide de l'équation d'état (2.4). La correction : on corrige la prédiction en tenant compte de la nouvelle observation Y_k et de l'équation d'observation (2.5) afin d'obtenir π_k .

Un point important est de déterminer ce que la nouvelle observation Y_k apporte de nouveau par rapport aux observations passées $Y_{1:k-1}$, la réponse est donnée par le processus innovation (cf. [11]) :

$$\nu_k \stackrel{\text{déf}}{=} Y_k - \hat{Y}_{k-} \quad \text{où} \quad \hat{Y}_{k-} \stackrel{\text{déf}}{=} \mathbb{E}[Y_k | Y_{1:k-1}].$$

Proposition 2.2.1 *L'innovation s'exprime de la façon suivante :*

$$\nu_k = Y_k - H_k \hat{X}_{k-}. \quad (2.6)$$

Elle forme une suite indépendante de variables aléatoires de loi $\nu_k \sim \mathcal{N}(0, Q_k^\nu)$ avec

$$Q_k^\nu = H_k R_{k-}^* H_k^* + Q_k^V \quad (2.7)$$

et $\nu_k \perp Y_{1:k-1}$.

Ainsi les informations contenues dans $(Y_{1:k-1}, Y_k)$ et dans $(Y_{1:k-1}, \nu_k)$ sont identiques.

Proposition 2.2.2 *la loi conditionnelle π_k est gaussienne d'espérance \hat{X}_k et de matrice de covariance R_k . Ces deux quantités sont calculées récursivement à l'aide du filtre de Kalman :*

Initialisation :

$$\hat{X}_0 = \bar{X}_0, \quad R_0 = Q_0,$$

Prédiction :

$$\hat{X}_{k^-} = F_{k-1} \hat{X}_{k-1}, \quad (2.8)$$

$$R_{k^-} = F_{k-1} R_{k-1} F_{k-1}^* + G_{k-1} Q_{k-1}^W G_{k-1}^*, \quad (2.9)$$

Correction :

$$K_k = R_{k^-} H_k^* [H_k R_{k^-} H_k^* + Q_k^V]^{-1}, \quad (2.10)$$

$$\hat{X}_k = \hat{X}_{k^-} + K_k [Y_k - H_k \hat{X}_{k^-}], \quad (2.11)$$

$$R_k = [I - K_k H_k] R_{k^-}. \quad (2.12)$$

K_k est appelé gain de Kalman.

Preuve L'initialisation est évidente, passons à la démonstration de la prédiction. Comme le vecteur $(X_k, Y_{1:k-1})$ est gaussien, la loi conditionnelle de X_k sachant $Y_{1:k-1}$ est également gaussienne, on note \hat{X}_{k^-} sa moyenne et R_{k^-} sa covariance. On a :

$$\begin{aligned} \hat{X}_{k^-} &= \mathbb{E}[X_k | Y_{1:k-1}] \\ &= F_{k-1} \mathbb{E}[X_{k-1} | Y_{1:k-1}] + G_{k-1} \mathbb{E}[W_{k-1} | Y_{1:k-1}]. \end{aligned}$$

Comme $\mathbb{E}[X_{k-1} | Y_{1:k-1}] = \hat{X}_{k-1}$ et $\mathbb{E}[W_{k-1} | Y_{1:k-1}] = 0$ on en déduit (2.8) d'une part et

$$X_k - \hat{X}_{k^-} = F_{k-1} (X_{k-1} - \hat{X}_{k-1}) + G_{k-1} W_{k-1}$$

d'autre part, d'où

$$\begin{aligned} R_{k^-} &= \mathbb{E}[(X_k - \hat{X}_{k^-})(X_k - \hat{X}_{k^-})^*] \\ &= \mathbb{E}[(F_{k-1} (X_{k-1} - \hat{X}_{k-1}) + G_{k-1} W_{k-1}) (")^*] \\ &= F_{k-1} R_{k-1} F_{k-1}^* + G_{k-1} Q_{k-1}^W G_{k-1}^*. \end{aligned}$$

En effet $(X_k - \hat{X}_k) \perp W_k$ donc $\mathbb{E}[(X_{k-1} - \hat{X}_{k-1}) W_{k-1}^*] = 0$. Ce qui prouve (2.9).

Passons à la démonstration de l'étape de correction. Le vecteur $(X_k, Y_{1:k})$ est gaussien, la loi conditionnelle de X_k sachant $Y_{1:k}$ est également gaussienne, on note \hat{X}_k sa moyenne et R_k sa covariance. On a :

$$\hat{X}_k = \mathbb{E}[X_k | Y_{1:k}] = \hat{X}_{k-} + \mathbb{E}[X_k - \hat{X}_{k-} | Y_{1:k}].$$

D'après la définition de l'innovation (proposition 2.2.1), on a que :

$$\mathbb{E}[X_k - \hat{X}_{k-} | Y_{1:k}] = \mathbb{E}[X_k - \hat{X}_{k-} | Y_{1:k-1}, \nu_k]$$

et comme $Y_{1:k-1} \perp (X_k - \hat{X}_{k-}, \nu_k)$, on obtient que :

$$\mathbb{E}[X_k - \hat{X}_{k-} | Y_{1:k-1}, \nu_k] = \mathbb{E}[X_k - \hat{X}_{k-} | \nu_k].$$

Ainsi

$$\hat{X}_k = \hat{X}_{k-} + \mathbb{E}[X_k - \hat{X}_{k-} | \nu_k]$$

et

$$\begin{aligned} X_k - \hat{X}_k &= (X_k - \hat{X}_{k-}) - (\hat{X}_k - \hat{X}_{k-}) \\ &= (X_k - \hat{X}_{k-}) - \mathbb{E}[X_k - \hat{X}_{k-} | \nu_k]. \end{aligned}$$

On calcule maintenant la moyenne et covariance de $(\hat{X}_k - \hat{X}_{k-}) | \nu_k$. Le vecteur $(X_k - \hat{X}_{k-}, \nu_k)$ est gaussien, centré de covariance

$$\begin{pmatrix} R_{k-} & R_{k-} H_k^* \\ H_k R_{k-} & H_k R_{k-} H_k^* + Q_k^V \end{pmatrix}.$$

On a donc que la loi conditionnelle de $X_k - \hat{X}_{k-}$ sachant ν_k est gaussienne dont la moyenne et la covariance sont données par :

$$\begin{aligned} \hat{X}_k &= \hat{X}_{k-} + R_{k-} H_k^* [H_k R_{k-} H_k^* + Q_k^V]^{-1} \nu_k, \\ R_k &= R_{k-} - R_{k-} H_k^* [H_k R_{k-} H_k^* + Q_k^V]^{-1} H_k R_{k-}. \end{aligned}$$

□

Précisons que le filtre de Kalman est à la fois un théorème et un algorithme.

Algorithme 2.2.3 (Filtre de Kalman)

```

%initialisation  $\hat{X}_0 \leftarrow \bar{X}_0$ 
 $R_0 \leftarrow Q_0$ 
%itérations
pour  $i = 1, 2, \dots$  faire
  %prédiction
     $\hat{X}_k^- \leftarrow F_{k-1} \hat{X}_{k-1}$ 
     $R_{k-} \leftarrow F_{k-1} R_{k-1} F_{k-1}^* + G_{k-1} Q_{k-1}^W G_{k-1}^*$ 
  %correction
     $K_k \leftarrow R_{k-} H_k^* [H_k R_{k-} H_k^* + Q_k^V]^{-1}$ 
     $\hat{X}_k \leftarrow \hat{X}_{k-} + K_k [Y_k - H_k \hat{X}_{k-}]$ 
     $R_k \leftarrow [I - K_k H_k] R_{k-}$ 
fin pour

```

Le filtre de Kalman étant optimal dans le cas d'un modèle linéaire gaussien, nous allons étudier un exemple qui nous servira par la suite d'exemple de référence :

Exemple 2.2.4 Soit un modèle linéaire gaussien :

$$\begin{aligned}
 X_{k+1} &= A X_k + \begin{pmatrix} 3 \\ 3 \end{pmatrix} + B W, & \forall k \geq 0, \\
 Y_k &= C X_k + V, & \forall k \geq 1,
 \end{aligned}$$

où $A = \begin{pmatrix} 0.2 & 0.2 \\ 0.5 & 0.5 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $C = (1 \ 1)$,

$X_0 \sim \mathcal{N}(0, I)$, $W \sim \mathcal{N}(0, \sigma_W^2 I)$, $V \sim \mathcal{N}(0, \sigma_V^2 I)$ et X_0, W, V sont indépendants. Cet exemple permet de visualiser la "vraie" trajectoire, l'estimation de la trajectoire par le filtre de Kalman et, à certains moments, les ellipses de confiance à 90% (cf. A.1). Les ellipses de confiance généralisent les intervalles de confiance à \mathbb{R}^2 . Elles permettent d'avoir une "zone" de confiance. Deux cas se présentent, intensité du bruit d'observation petite et grande :

$\sigma_V^2 = 0.02$: la trajectoire estimée est très proche de la trajectoire réelle et la présence de notre approximation dans la zone de confiance confirme la bonne qualité de l'estimation (cf. figure 2.1).

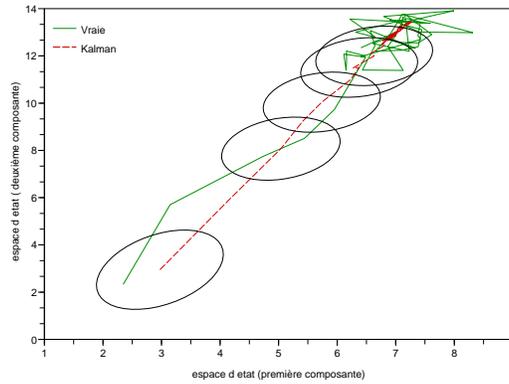
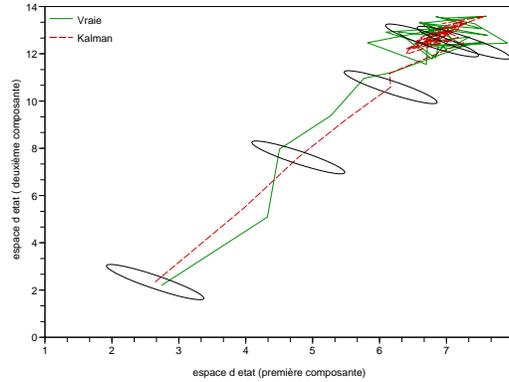


FIG. 2.1 – Filtre de Kalman et ellipses de confiance pour $T = 50$, $\sigma_V^2 = 0.02$ et $\sigma_W^2 = 0.2$

FIG. 2.2 – Filtre de Kalman et ellipses de confiance pour $T = 50$, $\sigma_V^2 = 8$ et $\sigma_W^2 = 0.2$

$\sigma_V^2 = 8$: l'approximation est un peu moins précise, car l'intensité du bruit d'observation est grande. Elle reste tout de même de bonne qualité au vue des ellipses de confiance (cf. figure 2.2).

Ces deux cas mettent en évidence la bonne qualité de l'estimation de la trajectoire par le filtre de Kalman. Les ellipses de confiance sont naturellement plus grandes que l'intensité du bruit d'observation est grande.

2.2.2 Filtre de kalman étendu

Considérons le système non linéaire décrit à l'exemple 1.2.2 :

$$X_{k+1} = f(X_k) + g(X_k) W_k, \quad (2.13)$$

$$Y_k = h(X_k) + V_k. \quad (2.14)$$

On suppose que $\mathbb{E}[W_k] = 0$, $cov[W_k] = Q_k^W$, $\mathbb{E}[V_k] = 0$, $cov[V_k] = Q_k^V$ et que W_k, V_k, X_0 sont indépendants. Le filtre optimal décrit dans la proposition 2.1.1 ne peut se résoudre explicitement comme dans le cas d'un modèle linéaire / gaussien vu dans la section précédente. Une idée naturelle est alors de linéariser le système autour de l'estimée courante et d'appliquer la technique du filtre de Kalman. Pour l'initialisation on pose :

$$\hat{X}_0 = \mathbb{E}[X_0], \quad R_0 = cov(X_0). \quad (2.15)$$

où X_0 n'est pas nécessairement gaussien. Pour la phase de prédiction on linéarise l'équation d'état autour de \hat{X}_{k-1} :

$$X_k = f(X_{k-1}) + g(X_{k-1}) W_{k-1} \quad (2.16)$$

$$\simeq f(\hat{X}_{k-1}) + F_{k-1} (X_{k-1} - \hat{X}_{k-1}) + G_{k-1} W_{k-1}. \quad (2.17)$$

où

$$F_{k-1} \stackrel{\text{déf}}{=} \nabla f(\hat{X}_{k-1}), \quad G_{k-1} \stackrel{\text{déf}}{=} g(\hat{X}_{k-1}).$$

Ainsi

$$\begin{aligned} \hat{X}_{k-} &= \mathbb{E}[X_k | Y_{1:k-1}] \\ &\simeq \mathbb{E}[f(\hat{X}_{k-1}) | Y_{1:k-1}] + F_{k-1} \mathbb{E}[X_{k-1} - \hat{X}_{k-1} | Y_{1:k-1}] + G_{k-1} \mathbb{E}[W_{k-1} - \hat{X}_{k-1} | Y_{1:k-1}] \\ &= f(\hat{X}_{k-1}). \end{aligned}$$

La covariance prédite est donnée par :

$$\begin{aligned} R_{k-} &= \mathbb{E}[(X_k - \hat{X}_{k-})(X_k - \hat{X}_{k-})^*] \\ &\simeq F_{k-1}^* R_{k-1} F_{k-1} + G_{k-1} Q_{k-1}^W G_{k-1}^*. \end{aligned}$$

Pour la phase de correction on linéarise l'équation d'observation autour de l'estimée, c'est à dire X_{k-} , on obtient :

$$\begin{aligned} Y_k &= h(X_k) + V_k, \\ &\simeq h(\hat{X}_{k-}) + H_k (X_k - \hat{X}_{k-}) + V_k. \end{aligned}$$

En appliquant le filtre de Kalman on obtient :

$$\begin{aligned} K_k &= R_{k^-} H_k^* [H_k R_{k^-} H_k^* + Q_k^V]^{-1} \\ \hat{X}_k &= \hat{X}_{k^-} + K_k [Y_k - h(\hat{X}_{k^-})], \\ R_k &= [I - K_k H_k] R_{k^-}. \end{aligned}$$

Algorithme 2.2.5 (Filtre de kalman étendu)

$\hat{X}_0 \leftarrow \bar{X}_0$
 $R_0 \leftarrow Q_0$
pour $i = 1, 2, \dots$ **faire**
 $F_{k-1} \leftarrow \nabla f(\hat{X}_{k-1})$
 $\hat{X}_k^- \leftarrow F_{k-1} \hat{X}_{k-1}$
 $R_{k^-} \leftarrow F_{k-1} R_{k-1} F_{k-1}^* + G_{k-1} Q_{k-1}^W G_{k-1}^*$
 $H_k \leftarrow \nabla h(\hat{X}_{k^-})$
 $K_k \leftarrow R_{k^-} H_k^* [H_k R_{k^-} H_k^* + Q_k^V]^{-1}$
 $\hat{X}_k \leftarrow \hat{X}_{k^-} + K_k [Y_k - h(\hat{X}_{k^-})]$
 $R_k \leftarrow [I - K_k H_k] R_{k^-}$
fin pour

Exemple 2.2.6 (Filtre de Kalman étendu : cas non linéaire simple)

Considérons un modèle non linéaire simple :

$$\begin{aligned} X_{k+1} &= f(X_k) + W_k, \\ Y_k &= h(X_k) + V_k. \end{aligned}$$

où

$$f \begin{pmatrix} X_k^1 \\ X_k^2 \end{pmatrix} = \begin{pmatrix} \cos(X_k^1) \\ \frac{X_k^1 X_k^2}{5} \end{pmatrix}, \quad h \begin{pmatrix} X_k^1 \\ X_k^2 \end{pmatrix} = \begin{pmatrix} X_k^1 \\ X_k^1 X_k^2 \end{pmatrix},$$

$X_0 \sim \mathcal{N}(0, I)$, $W \sim \mathcal{N}(0, \sigma_W^2 I)$, $V \sim \mathcal{N}(0, \sigma_V^2 I)$ et X_0, W, V sont indépendants. Nous estimons la qualité de notre estimation en calculant l'intervalle de confiance par approximation gaussienne. On utilisera l'espérance et la covariance obtenues par le filtre de Kalman étendu. On représente sur chaque graphique la trajectoire réelle, l'estimation de la trajectoire par le filtre de

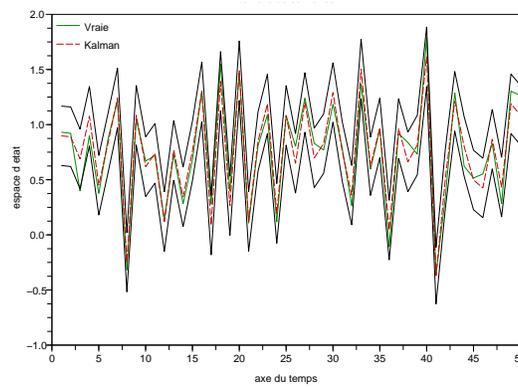


FIG. 2.3 – Filtre de Kalman étendu et approximation gaussienne pour $T = 50$, $\sigma_V^2 = 0.02$ et $\sigma_W^2 = 1$

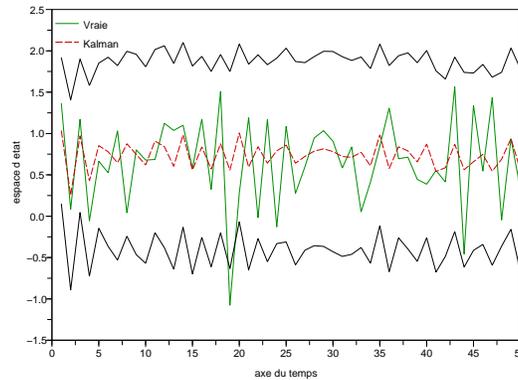


FIG. 2.4 – Filtre de Kalman étendu et approximation gaussienne pour $T = 50$, $\sigma_V^2 = 8$ et $\sigma_W^2 = 1$

Kalman étendu et l'intervalle de confiance calculé par approximation gaussienne. Comme dans l'exemple 2.2.4 deux cas se présentent, intensité du bruit d'observation petite et grande :

$\sigma_V^2 = 0.02$: la trajectoire estimée et la trajectoire réelle sont très proches. L'intervalle de confiance confirme la qualité de notre estimation (cf. figure 2.3).

$\sigma_V^2 = 8$: l'intervalle de confiance est beaucoup plus grand ce qui montre une

approximation moins précise (cf. figure 2.4).

Le filtre de Kalman étendu fournit donc, pour ce modèle non linéaire simple, une bonne approximation.

Chapitre 3

Filtrage particulaire

Les méthodes particulières sont fondées sur le principe de Monte Carlo pour estimer et prédire en ligne des systèmes dynamiques. L'idée est d'approcher la loi π_k à l'aide d'une mesure discrète égale à une somme finie de mesures de Dirac concentrées en des points dénommés "particules" et pondérées par des coefficients appelés "poids des particules". L'approximation consiste donc à écrire le filtre prédit et le filtre sous la forme :

$$\begin{aligned}\pi_{k^-}(\mathrm{d}x) &\simeq \pi_{k^-}^N(\mathrm{d}x) = \sum_{i=1}^N w_{k^-}^i \delta_{\xi_{k^-}^i}(\mathrm{d}x), \\ \pi_k(\mathrm{d}x) &\simeq \pi_k^N(\mathrm{d}x) = \sum_{i=1}^N w_k^i \delta_{\xi_k^i}(\mathrm{d}x).\end{aligned}$$

Ces méthodes séquentielles de Monte Carlo (ou filtres particulières) font appel, de manière récursive, au principe d'échantillonnage d'importance et de rééchantillonnage. Nous conseillons la lecture de le livre de Doucet [6] pour faire le point sur les connaissances dans le domaine ainsi que l'article de Gordon [10] et le livre de Simon J. Godsill, Olivier Cappé, et Eric Moulines [9].

3.1 Filtre de Monte Carlo et Bootstrap

Filtre de Monte Carlo

Comme nous l'avons vu précédemment, on cherche à approximer le filtre de façon récursive, supposons que :

$$\pi_{k-1}(dx) = \sum_{i=1}^N w_{k-1}^i \delta_{\xi_{k-1}^i}(dx).$$

Algorithme 3.1.1 On obtient alors une approximation du filtre prédit et du filtre grâce aux équations (2.2) et (2.3) :

Prédiction :

$$\pi_{k-}(dx) \simeq \pi_{k-}^N(dx) = \sum_{i=1}^N w_{k-}^i \delta_{\xi_{k-}^i}(dx) \quad \text{où} \quad \begin{cases} \xi_{k-}^i \sim Q_k(\xi_{k-1}^i, dx) \\ w_{k-}^i = w_{k-1}^i \end{cases}$$

Correction :

$$\pi_k(dx) \simeq \pi_k^N(dx) = \sum_{i=1}^N w_k^i \delta_{\xi_k^i}(dx) \quad \text{où} \quad \begin{cases} \xi_k^i = \xi_{k-}^i \\ w_k^i \propto w_{k-1}^i \psi_k(\xi_{k-}^i, y_k) \end{cases}$$

Preuve :

Prédiction :

En utilisant l'équation de prédiction (2.2), on obtient :

$$\begin{aligned} \pi_{k-}(dx) &= \int_{\mathbb{R}^n} Q_k(x', dx) \sum_{i=1}^N w_{k-1}^i \delta_{\xi_{k-1}^i}(dx') \\ &= \sum_{i=1}^N w_{k-1}^i Q_k(\xi_{k-1}^i, dx) \\ &\simeq \pi_{k-}^N(dx) = \sum_{i=1}^N w_{k-}^i \delta_{\xi_{k-}^i}(dx). \end{aligned}$$

où $\xi_{k-}^i \sim Q_k(\xi_{k-1}^i, dx)$ et $w_{k-}^i = w_{k-1}^i$.

Cette étape est également appelée étape de propagation des particules ; dans le cas précis du filtre de Monte Carlo la propagation se fait selon la loi de transition de la chaîne.

Correction :

L'équation de correction (2.3) donne :

$$\begin{aligned}
\pi_k(\mathrm{d}x) &= \frac{\psi_k(x, y_k) \pi_{k-}(\mathrm{d}x)}{\int_{\mathbb{R}^n} \psi_k(x', y_k) \pi_{k-}(\mathrm{d}x')} \\
&\simeq \pi_k^N(\mathrm{d}x) = \frac{\psi_k(x, y_k) \sum_{i=1}^N w_{k-1}^i \delta_{\xi_{k-}^i}(\mathrm{d}x)}{\int_{\mathbb{R}^n} \psi_k(x', y_k) \sum_{i=1}^N w_{k-1}^i \delta_{\xi_{k-}^i}(\mathrm{d}x')} \\
&= \frac{\sum_{i=1}^N w_{k-1}^i \psi_k(\xi_{k-}^i, y_k) \delta_{\xi_{k-}^i}(\mathrm{d}x)}{\sum_{i=1}^N \psi_k(\xi_{k-}^i, y_k) w_{k-1}^i} \\
&= \sum_{i=1}^N w_k^i \delta_{\xi_k^i}(\mathrm{d}x).
\end{aligned}$$

où $w_k^i = \frac{w_{k-1}^i \psi_k(\xi_{k-}^i, y_k)}{\sum_{i=1}^N w_{k-1}^i \psi_k(\xi_{k-}^i, y_k)}$ et $\xi_k^i = \xi_{k-}^i$. Cette étape effectue une mise à jour des particules.

Algorithme 3.1.2 (*Filtre de Monte Carlo*)

```

 $\xi_{1:N} \sim \mu_0$ 
 $w_{1:N} \leftarrow 1/N$ 
pour  $k = 2 : T$  faire
     $\tilde{\xi}_i \sim Q_k(\xi_i, \cdot)$  pour  $i = 1 : N$  %propagation
     $\tilde{w}_i \leftarrow w_i \psi_k(\tilde{\xi}_i, y_k)$  pour  $i = 1 : N$  %mise à jour des poids
     $w_i \leftarrow \tilde{w}_i / \sum_{i=1}^N \tilde{w}_i$  pour  $i = 1 : N$  %normalisation
     $\xi_{1:N} \leftarrow \tilde{\xi}_{1:N}$ 
sortie  $(\xi_{1:N}, w_{1:N})$ 
fin pour

```

Ce filtre n'est pas viable dans la pratique car il s'avère que la majorité des poids dégénère très rapidement vers zéro or idéalement tous les poids devraient rester proches de $1/N$, ce qui signifierait que les particules sont d'égale importance dans l'approximation. Ce filtre ne devient donc utilisable que lorsqu'une étape de rééchantillonnage est ajoutée permettant ainsi de limiter

la dégénérescence des poids. En effet, cette étape va favoriser les particules dites “importantes” par rapport aux particules ayant un poids négligeable, de ce fait on duplique les particules à fort poids au détriment de celles qu’on souhaite défavoriser. On obtient un des premiers algorithmes proposé en 93 par Gordon, Salmond et Smith [10], ce filtre est connu sous le nom de *filtre Bootstrap*. Nous conserverons dans la suite la dénomination de filtre Bootstrap pour désigner cette classe initiale de filtres particulières, pour laquelle les particules sont propagées selon la loi de transition et rééchantillonnées à chaque étape selon un schéma classique de rééchantillonnage. L’exemple qui va suivre va nous permettre deux choses : de constater la dégénérescence rapide du filtre de Monte Carlo et de voir si une approximation gaussienne a du “sens”. En effet, on va tracer l’intervalle de confiance empirique et l’intervalle de confiance calculé par approximation gaussienne, la ressemblance des deux intervalles nous permettra d’émettre une hypothèse de normalité.

Exemple 3.1.3 (Filtre de Monte Carlo pour un cas non linéaire simple)

Considérons le modèle non linéaire simple de l’exemple 2.2.6 muni des mêmes hypothèses. On représente sur chaque graphique la trajectoire approchée par le filtre de Monte Carlo, la trajectoire réelle et l’intervalle de confiance. Deux cas se présentent :

- *intervalle de confiance empirique (cf. figure 3.1)*
- *intervalle de confiance calculé par approximation gaussienne (cf. figure 3.2)*

On peut dire deux choses : l’hypothèse de normalité peut être émise (similarité des intervalles) et l’approximation faite par le filtre de Monte Carlo est assez mauvaise. En effet, la trajectoire réelle ne se trouve que très rarement dans l’intervalle de confiance. Ces deux graphiques confirment la dégénérescence, en quelques itérations, des méthodes séquentielles de Monte Carlo sans rééchantillonnage.

Rééchantillonnage

Avant d’exposer l’algorithme du bootstrap intéressons-nous aux différentes techniques de rééchantillonnage (ou redistribution des particules) que nous utiliserons par la suite.

En toute logique il faudrait utiliser une technique de rééchantillonnage multinomial :

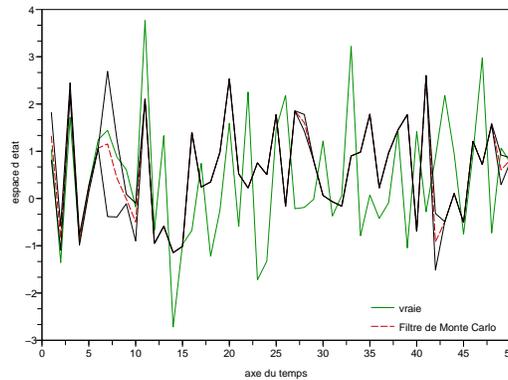
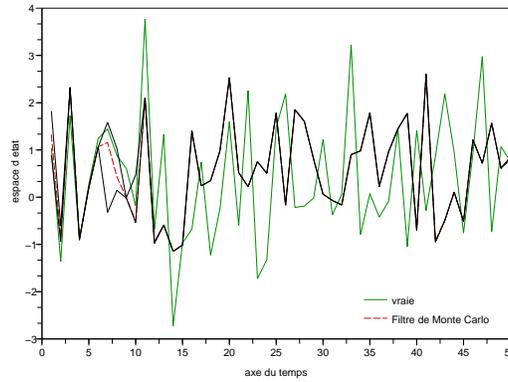


FIG. 3.1 – Filtre de Monte Carlo et intervalle de confiance empirique pour $T = 50$, $N = 1000$, $\sigma_V^2 = 0.1$ et $\sigma_W^2 = 1$

FIG. 3.2 – Filtre de Monte Carlo et intervalle de confiance calculé par approximation gaussienne pour $T = 50$, $N = 1000$, $\sigma_V^2 = 0.1$ et $\sigma_W^2 = 1$

Algorithme 3.1.4 (Resample)

procédure *resample_multi*($w_{1:N}, x_{1:N}$)

pour $i = 1 : N$ *faire*

$u \sim \mathcal{U}[0, 1]$

$j \leftarrow 1$

tant que $w_1 + w_2 + \dots + w_j < u$ *faire*

$j \leftarrow j + 1$

fin tant que

$\xi_i \leftarrow x_j$

fin pour

rendre $\xi_{1:N}$

Cet algorithme consiste à tirer N particules parmi $\{\xi_k^1, \dots, \xi_k^N\}$ avec les probabilités $\{w_k^1, \dots, w_k^N\}$, on obtient ainsi j_1 fois la particule ξ_k^1 , j_2 fois la particule ξ_k^2, \dots . Le vecteur (j_1, \dots, j_N) suit la loi multinomiale suivante $\mathcal{M}(N, w_k^1, \dots, w_k^N)$. La complexité de cet algorithme est en $\mathcal{O}(N \log(N))$, il peut être amélioré en triant au préalable les poids par ordre croissant. D'autre part, en faisant appel aux statistiques d'ordre, on passe à une complexité en $\mathcal{O}(N)$.

Une autre technique de redistribution des particules est la redistribution des résidus (cf. A.2), on sélectionne dans un premier temps pour chaque i la particule ξ^i de façon déterministe $\lfloor N w^i \rfloor$ fois, il reste donc $N - \sum_{i=1}^N \lfloor N w^i \rfloor$ particules à sélectionner, on applique alors la méthode de redistribution multinomiale à la loi de probabilité correspondante aux poids restants : $\tilde{w}^i \leftarrow w^i - (1/N) \lfloor N w^i \rfloor$ et $\tilde{w}^i \leftarrow \tilde{w}^i / \sum_{j=1}^N \tilde{w}^j$.

Notons tout de même que la technique de redistribution des résidus est plus rapide que la technique du rééchantillonnage multinomial. Cependant le rééchantillonnage multinomial est le genre de technique qui est utilisée en pratique.

Bootstrap

Algorithme 3.1.5 (*Filtre Bootstrap*)

```

 $\xi_{1:N} \sim \mu_0$ 
 $w_{1:N} \leftarrow 1/N$ 
pour  $k = 2 : T$  faire
     $\tilde{\xi}_i \sim Q_k(\xi_i, \cdot)$  pour  $i = 1 : N$  %propagation
     $\tilde{w}_i \leftarrow \psi_k(\tilde{\xi}_i, y_k)$  pour  $i = 1 : N$  %mise à jour des poids
     $w_i \leftarrow \tilde{w}_i / \sum_{i=1}^N \tilde{w}_i$  pour  $i = 1 : N$  %normalisation
     $\xi_{1:N} \leftarrow \text{resample\_multi}(w_{1:N}, \tilde{\xi}_{1:N})$ 
    sortie  $(\xi_{1:N})$ 
fin pour

```

Comme nous venons de le voir, l'algorithme du bootstrap est identique à l'algorithme du filtre de Monte Carlo seule une étape de rééchantillonnage est ajoutée. Nous appellerons *resample* une technique quelconque de rééchantillonnage. Pour nous rendre compte de l'amélioration apportée par cette

étape, étudions un exemple basé sur le système non linéaire simple utilisé dans l'exemple 3.1.3.

Exemple 3.1.6 (Bootstrap : cas non linéaire simple) *Cet exemple permet de visualiser la trajectoire réelle, la trajectoire estimée par le filtre bootstrap et l'intervalle de confiance calculé de deux façons différentes :*

- *intervalle de confiance empirique (cf. figure 3.3)*
- *intervalle de confiance calculé par approximation gaussienne (cf. figure 3.5)*

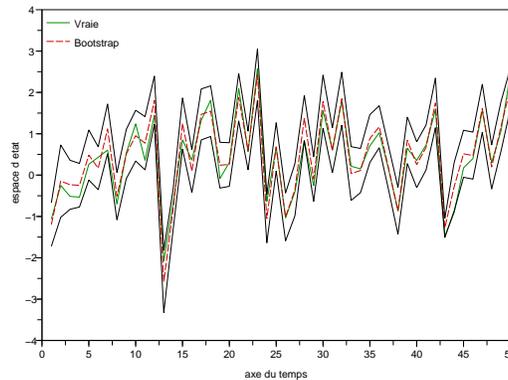
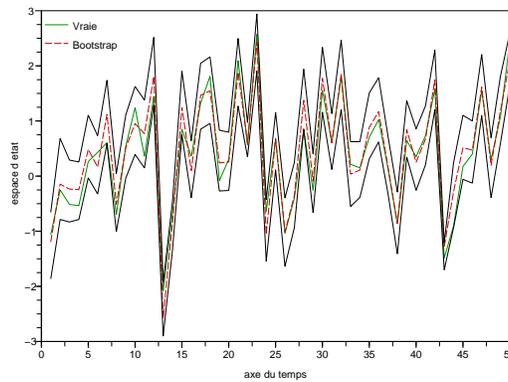


FIG. 3.3 – Filtre bootstrap et intervalle de confiance empirique pour $T = 50$, $N = 1000$, $\sigma_V^2 = 0.1$ et $\sigma_W^2 = 1$

FIG. 3.4 – Filtre bootstrap et intervalle de confiance calculé par approximation gaussienne pour $T = 50$, $N = 1000$, $\sigma_V^2 = 0.1$ et $\sigma_W^2 = 1$

Ils permettent de dire deux choses. D'une part que l'estimation fournie

par le filtre bootstrap est bonne, en effet la vraie trajectoire se trouve dans l'intervalle de confiance et la variance reste raisonnable, ceci est en partie dû à l'étape de rééchantillonnage et d'autre part que l'hypothèse de normalité peut être émise (similarité des intervalles de confiance).

En comparant avec les résultats obtenus par le filtre de Monte Carlo on constate que le rééchantillonnage est nécessaire pour obtenir une bonne approximation. Dans la pratique, pour mettre en œuvre cet algorithme on doit savoir :

- (i) Simuler selon μ_0 ,
- (ii) Simuler selon $Q_k(\xi, \cdot)$ pour tout ξ ,
- (iii) Calculer la vraisemblance $\psi_k(y_k, x_k)$ pour tout x_k et de disposer d'une routine de redistribution.

Le problème de cet algorithme est qu'il est intimement lié à la capacité de la loi de transition à prédire les particules dans les régions de forte vraisemblance. On peut avoir deux types de configurations "critiques" : dans le cas où le modèle de mesure est très précis et où l'a priori est peu informatif peu de particules auront un poids non négligeable par conséquent lors du rééchantillonnage la majorité des particules sera éliminée et les rares autres seront dupliquées un grand nombre de fois, on aura un phénomène d'appauvrissement des états. Le deuxième cas de figure est lorsque le support de la loi de transition est relativement distinct de la vraisemblance, les particules auront toutes une faible vraisemblance et, après rééchantillonnage, les poids des particules seront à peu près tous égaux, leur diversité sera donc conservée mais aucune ne correspondra réellement à une bonne hypothèse sur le vecteur d'état, on aura un phénomène de dégénérescence des particules. Notre but va donc être de "guider" nos particules dans des régions de forte vraisemblance, à l'aide d'une fonction, appelée fonction de proposition (ou d'importance) autre que la loi de transition. Cette étude nous amène à la section suivante.

Avant de passer à la description des algorithmes des filtres SIS et SIR, notons que certains auteurs ne rééchantillonnent pas systématiquement, en effet le critère empirique suivant :

$$N_k^{\text{eff}} = \frac{1}{\sum_{i=1}^N \omega_k^i} \in [0, 1]^N$$

qui représente le nombre efficace de particules, permet de justifier d'une étape de rééchantillonnage ou non en se fixant un taux. Plus précisément, on ne

rééchantillonne les particules que si N_k^{eff} est inférieur à un certain seuil N_{seuil} . Dans la suite de notre exposé nous effectuerons un rééchantillonnage systématiquement à chaque pas de temps.

3.2 Filtres SIS et SIR

Les filtres SIS (Sequential Importance Sampling) et SIR (Sequential Importance Resampling) correspondent respectivement au filtre de Monte Carlo et au filtre bootstrap dans lesquels l'étape de propagation des particules s'effectue non plus au travers de la loi de transition de la chaîne, mais grâce à une loi de proposition.

SIR : Filtre particulière à échantillonnage pondéré séquentiel avec rééchantillonnage

Comme nous avons pu l'expliquer dans le paragraphe précédent, la loi de proposition utilisée dans le filtre SIR devrait permettre de propager les particules dans des régions de forte vraisemblance. Essayons de "mettre en place" cet algorithme. Pour ceci revenons à l'étape de prédiction du filtre de Monte Carlo (cf. la preuve de la proposition 3.1.1) :

$$\begin{aligned}\pi_{k-}(\mathrm{d}x) &= \int_{\mathbb{R}^n} Q_k(x', \mathrm{d}x) \sum_{i=1}^N w_{k-1}^i \delta_{\xi_{k-1}^i}(\mathrm{d}x') \\ &= \sum_{i=1}^N w_{k-1}^i Q_k(\xi_{k-1}^i, \mathrm{d}x).\end{aligned}$$

Supposons que l'on se donne un noyau de proposition de la forme $\tilde{Q}_k(\mathrm{d}x, y, x')$. Ainsi :

$$\begin{aligned}\pi_{k-}(\mathrm{d}x) &= \sum_{i=1}^N \frac{w_{k-1}^i Q_k(\xi_{k-1}^i, \mathrm{d}x) \tilde{Q}_k(\mathrm{d}x, y_k, \xi_{k-1}^i)}{\tilde{Q}_k(\mathrm{d}x, y_k, \xi_{k-1}^i)} \\ &\simeq \pi_{k-}^N(\mathrm{d}x) = \sum_{i=1}^N w_{k-1}^i \frac{Q_k(\xi_{k-1}^i, \mathrm{d}x)}{\tilde{Q}_k(\mathrm{d}x, y_k, \xi_{k-1}^i)} \delta_{\xi_{k-}^i}(\mathrm{d}x).\end{aligned}$$

où $\xi_{k-}^i \sim \tilde{Q}_k(\mathrm{d}x, y_k, \xi_{k-1}^i)$.

Ce noyau prend donc en compte la nouvelle observation, d'autre part nous devons faire une hypothèse de domination. En effet, on doit imposer l'absolu continuité de la loi de transition par rapport au noyau de proposition \tilde{Q}_k , i.e. $Q_k \ll \tilde{Q}_k$:

$$Q_k(x', dx) = f_k(x, y, x') \tilde{Q}_k(dx, y_k, x') \quad \forall k \geq 1.$$

On obtient donc que :

$$\pi_k^N(dx) = \sum_{i=1}^N w_{k-1}^i f_k(x, y_k, \xi_{k-1}^i) \delta_{\xi_{k-}^i}(dx).$$

En appliquant l'équation de correction (2.3) on a :

$$\begin{aligned} \pi_k(dx) &\simeq \pi_k^N(dx) \\ &= \frac{\psi_k(x, y_k) \sum_{i=1}^N w_{k-1}^i f_k(x, y_k, \xi_{k-1}^i) \delta_{\xi_{k-}^i}(dx)}{\int_{\mathbb{R}^n} \psi_k(x', y_k) \sum_{i=1}^N w_{k-1}^i f_k(x', y_k, \xi_{k-1}^i) \delta_{\xi_{k-}^i}(dx')} \\ &= \frac{\sum_{i=1}^N w_{k-1}^i \psi_k(\xi_{k-}^i, y_k) f_k(\xi_{k-}^i, y_k, \xi_{k-1}^i) \delta_{\xi_{k-}^i}(dx)}{\sum_{i=1}^N w_{k-1}^i \psi_k(\xi_{k-}^i, y_k) f_k(\xi_{k-}^i, y_k, \xi_{k-1}^i)} \\ &= \sum_{i=1}^N w_k^i \delta_{\xi_k^i}(dx). \end{aligned}$$

ainsi

$$\begin{aligned} \tilde{w}_k^i &= w_{k-1}^i \psi_k(\xi_{k-}^i, y_k) f_k(\xi_{k-}^i, y_k, \xi_{k-1}^i), \\ w_k^i &= \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i} \text{ et } \xi_k^i = \xi_{k-}^i. \end{aligned}$$

L'article de Doucet [5], propose une fonction d'importance optimale, dans le sens où elle minimise la variance des poids. La section suivante portera principalement sur l'amélioration du filtrage particulaire par différentes méthodes afin de limiter la dégénérescence des états, des poids et de minimiser la variance.

3.3 Améliorations

Ce paragraphe donne un aperçu des différentes méthodes existantes dans la littérature pour améliorer la technique de base du filtrage particulaire avec interactions, à savoir le filtre bootstrap. Ces différentes méthodes n'ont pas été utilisées en pratique par la communauté du «pistage» : elles ne sont pas assez rapides pour faire des calculs en temps réel. Dans le domaine de l'environnement, le contexte est un peu différent car il n'y a pas de contraintes de temps réel pour les calculs. Par conséquent, il peut être intéressant d'adapter ces méthodes.

3.3.1 Filtre particulaire auxiliaire (ASIR)

Lorsque l'on souhaite tirer des échantillons selon une loi de mélange, l'introduction de variables auxiliaires représentant la composante du mélange est classique. Shephard et Pitt [12] ont couplé cette idée avec le principe d'échantillonnage pondéré séquentiel dans le filtre particulaire auxiliaire, dénoté ASIR en anglais. Remarquons dans un premier temps que :

$$\begin{aligned} \pi_k(dx) &\propto \psi_k(x, y_k) \sum_{i=1}^N w_{k-1}^i Q_k(\xi_{k-1}^i, dx) \\ &\propto \sum_{i=1}^N \psi_k(x, y_k) w_{k-1}^i Q_k(\xi_{k-1}^i, dx), \\ &\propto \sum_{i=1}^N f_k(dx, i). \end{aligned}$$

Notre but est maintenant d'échantillonner selon cette loi jointe au moyen d'une loi de proposition g_k de la forme suivante :

$$g_k(dx, i|y_{1:k}) = g_k^1(i|y_{1:k}) g_k^2(dx|i, y_{1:k})$$

où

$$\begin{aligned} g_k^2(dx|i, y_{1:k}) &= Q_k(\xi_{k-1}^i, dx) \\ g_k^1(i|y_{1:k}) &\propto \psi_k(\mu_i^k, y_k) w_{k-1}^i. \end{aligned}$$

Le paramètre μ_i^k représente une caractéristique de la loi de transition : $Q_k(\xi_{k-1}^i, dx)$. Dans la suite on prendra $\mu_i^k = \mathbb{E}[X_k | X_{k-1} = \xi_{k-1}^i]$. Notre algorithme consiste donc à tirer N couples (ξ_k^j, i^j) selon cette loi. Ce tirage s'effectue dans un premier temps en tirant un indice i^j avec une probabilité $g_k^1(i | y_{1:k})$ puis à propager selon la loi de transition *a priori* la particule d'indice sélectionnée. Nous devons supposer l'absolue continuité de $\pi_k(dx, i)$ par rapport à $g_k(dx, i | y_{1:k})$, on a donc que :

$$\pi_k(dx, i) = \frac{\psi_k(x, y_k)}{\psi_k(\mu_i^k, y_k)} g_k(dx, i | y_{1:k}).$$

Calculons les poids mis à jour de ces particules :

$$\begin{aligned} \mathbb{E}[h(X_k) | Y_{1:k} = y_{1:k}] &= \int_{\mathbb{R}^n} h(x_k) \pi_k(dx_k) \\ &\propto \sum_{i=1}^N \int_{\mathbb{R}^n} h(x_k) \frac{f_k(dx_k, i)}{g(dx_k, i | y_{1:k})} g(dx_k, i | y_{1:k}) \\ &\propto \sum_{i=1}^N \int_{\mathbb{R}^n} h(x_k) \frac{\psi_k(x_k, y_k)}{\psi_k(\mu_i^k, y_k)} g(dx_k, i | y_{1:k}) \\ &\propto \mathbb{E}_{g_k(\cdot, i | y_{1:k})} [h(X_k) \frac{\psi_k(X_k, y_k)}{\psi_k(\mu_i^k, y_k)}] \\ &\simeq \sum_{j=1}^N h(\xi_k^j) \frac{\psi_k(\xi_k^j, y_k)}{\psi_k(\mu_{i^j}^k, y_k)}. \end{aligned}$$

On obtient donc que :

$$w_k^j \propto \frac{\psi_k(\xi_k^j, y_k)}{\psi_k(\mu_{i^j}^k, y_k)}.$$

Cet algorithme va donc nous permettre, grâce à ses deux étapes, de générer des particules qui ont plus de chance d'être dans une région de forte vraisemblance. Cet algorithme devient moins performant dès lors que le bruit du processus grandit, en effet le simple paramètre μ_i^k n'est pas suffisant pour caractériser la loi.

Algorithme 3.3.1 (ASIR)

$\xi_{1:N} \sim \mu_0$
 $w_{1:N} \leftarrow 1/N$
pour $k = 1 : T$ **faire**
 $\mu_i \leftarrow \mathbb{E}[X_k | X_{k-1} = \xi_i]$ pour $i = 1 : N$
 $\tilde{\beta}_i \leftarrow \psi_k(\mu_i, y_k) w_i$ pour $i = 1 : N$
 $\beta_i \leftarrow \tilde{\beta}_i / \sum_{i=1}^N \tilde{\beta}_i$ pour $i = 1 : N$ %normalisation
 $i^j \sim \text{multi}(\beta_i)$ pour $j = i : N$
 $\xi_j \sim Q_k(\xi_{i^j}, \cdot)$ pour $j = 1 : N$ %propagation
 $\tilde{w}_j \leftarrow \frac{\psi_k(\xi_j, y_k)}{\psi_k(\mu_{i^j}^k, y_k)}$ pour $j = 1 : N$ %mise à jour des poids
 $w_j \leftarrow \tilde{w}_j / \sum_{j=1}^N \tilde{w}_j$ pour $j = 1 : N$ %normalisation
sortie $(\xi_{1:N}, w_{1:N})$
fin pour

Exemple 3.3.2 Les commentaires sont similaires à l'exemple 3.1.6, on observe une bonne approximation par le filtre ASIR et l'hypothèse de normalité au vue de la ressemblance des graphiques peut être émise.

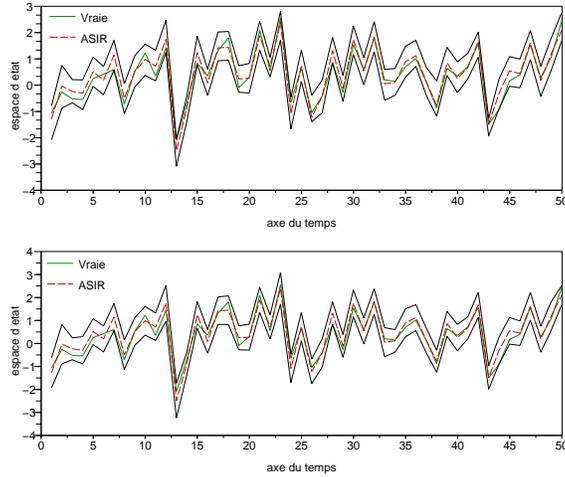


FIG. 3.5 – Filtre ASIR et intervalle de confiance calculé par approximation gaussienne (en haut) / Filtre ASIR et intervalle de confiance empirique (en bas) pour $T = 50$, $N = 100$, $\sigma_V^2 = 0.1$ et $\sigma_W^2 = 1$

Comme on peut le voir sur les exemples 3.3.2 et 3.1.6, les résultats des filtres bootstrap et ASIR sont équivalents. Pour les comparer plus précisément nous avons calculé la moyenne du MSE (moyenne quadratique des erreurs obtenue sur la trajectoire avec 5 répétitions) pour les deux filtres en fonction du nombre de particules. Cette comparaison est illustrée sur la figure 3.6. On voit ainsi que l'ASIR est moins variable que le filtre bootstrap. En contre-partie son algorithme est plus difficile à mettre en œuvre et son temps de calcul est plus élevé.

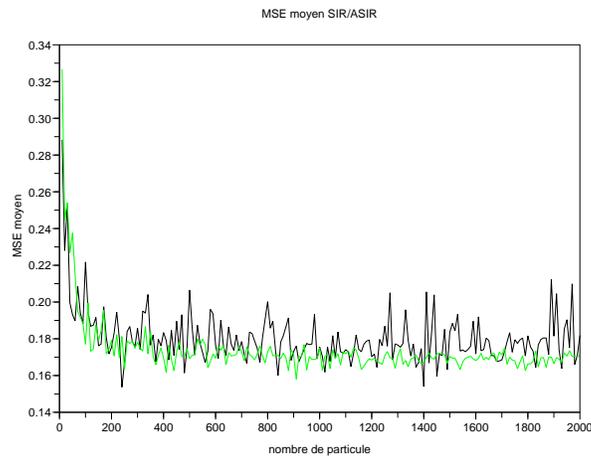


FIG. 3.6 – Évolution du MSE pour le filtre bootstrap (noir) et l'ASIR (clair) en fonction du nombre de particules

3.3.2 Méthodes de Monte Carlo par chaînes de Markov et filtres particuliers

Une idée “naturelle” reprise par de nombreux auteurs est d'utiliser les méthodes MCMC (Méthodes de Monte Carlo par Chaînes de Markov) pour propager nos particules. Berzuini et Gilks [8] parlent de *resample-move algorithm*. Rappelons que les méthodes MCMC ont deux inconvénients majeurs :

- Diagnostic de la convergence difficile

- Méthodes trop “coûteuses” pour être utilisables notamment dans le contexte d’une estimation séquentielle bayésienne.

MCMC et approximation de Laplace

Supposons que la loi conditionnelle π_k admette une densité par rapport à la mesure de Lebesgue :

$$\pi_k(dx) = p_k(x|y_{1:k}) dx, \quad \forall k \geq 0.$$

L’utilisation directe des méthodes MCMC dans le filtre particulière n’est pas possible. En effet, supposons que l’on dispose d’un échantillon $(\xi_{k-1}^i)_{i=1:N}$ de la loi conditionnelle π_{k-1} , une itération de l’algorithme de Metropolis-Hasting, qui est une méthode MCMC, consiste à proposer un candidat $\tilde{\xi}_i^k$ selon une loi de proposition $q_k(\cdot)$ pour tout $i = 1 : N$ et à accepter ce candidat avec la probabilité :

$$\alpha = \min\left(\frac{h_k(\tilde{\xi}_i^k)}{h_k(\xi_{k-1}^i)}, 1\right)$$

où

$$h_k(x) = \frac{\psi_k(x, y_k) p_k(x|y_{1:k-1})}{q_k(x)}.$$

La densité $p_k(x|y_{1:k-1})$ ne se calculant pas explicitement, les méthodes MCMC ne peuvent être utilisées directement. L’idée de Kass [14] pour “contourner” ce problème est la suivante. On remplace la densité inconnue $p_k(x|y_{1:k-1})$ par $\hat{p}_k(x|y_{1:k-1})$ tel que :

$$\hat{p}_k(x_k|y_{1:k-1}) = \int q_k(x_{k-1}, x_k) \hat{p}_{ost_{k-1}}(x_{k-1}|y_{1:k-1}) dx_{k-1}, \quad (3.1)$$

où $\hat{p}_{ost_{k-1}}(x_{k-1}|y_{1:k-1})$ est l’approximation gaussienne de $p_{k-1}(x_{k-1}|y_{1:k-1})$. En utilisant (3.1) l’algorithme de Metropolis-Hasting peut être utilisé, la fonction h_k devient :

$$h_k(x) = \frac{\psi_k(x, y_k) \hat{p}(x|y_{1:k-1})}{q_k(x)},$$

où $q_k(x) = \hat{p}_{ost_{k-1}}(x|y_{1:k-1})$.

Resample-move algorithm

Plaçons-nous dans le cas particulier où l'on cherche à estimer la loi d'un paramètre θ sachant une réalisation des observations $Y_{1:k} = y_{1:k}$ que l'on note $\pi_k(d\theta) = \mathbb{P}(d\theta | Y_{1:k} = y_{1:k})$, l'idée de l'algorithme dénommé *Resample-Move algorithm* introduit par Gilks et Berzuini est de combiner les méthodes MCMC, le rééchantillonnage et l'échantillonnage d'importance. Avant de le décrire, revenons sur quelques points. À k fixé, les poids $w_k(\theta) = \pi_{k+1}(d\theta)/\pi_k(d\theta)$ sont fonction de π_k et π_{k+1} , on notera $q_k(\theta, d\theta')$ le noyau de transition d'une chaîne de Markov qui a comme loi stationnaire π_{k+1} et vérifiant la propriété suivante :

$$\int_{\mathbb{R}^d} \pi_{k+1}(d\theta) q_k(\theta, d\theta') = \pi_{k+1}(d\theta'). \quad (3.2)$$

On notera S_k l'échantillon de particules à l'instant $t = k$ et n_k le nombre de particules générées, ainsi $S_k = \{\theta_k^{(j)}; j = 1, \dots, n_k\}$. Après l'initialisation, notre algorithme procède à une étape appelée *rejuvenation step*, elle se décompose en deux sous-étapes : rééchantillonnage et propagation des particules. Décrivons maintenant cet algorithme :

Algorithme 3.3.3 (*Resample-move algorithm*)

%initialisation

$t = 0$

$\theta_{1:n_0} \sim \pi_0$

%rejuvenation :

pour $k = 0, 1, \dots$ **faire**

%t = k + 1%

$w_i \leftarrow \pi_{k+1}(d\theta_i)/\pi_k(d\theta_i)$ pour $i = 1 : n_k$ *%mise à jour des poids*

$w_i \leftarrow w_i / \sum_{i=1}^{n_k} w_i$ pour $i = 1 : n_k$ *%normalisation*

$\tilde{\theta}_{1:n_{k+1}} \leftarrow \text{resample_multi}(w_{1:n_k}, \theta_{1:n_k})$ *%rééchantillonnage*

$w_i \leftarrow 1/n_{k+1}$

$\theta_i \sim q_k(\tilde{\theta}_i, \cdot)$ pour $i = 1 : n_{k+1}$ *%propagation*

fin pour

L'étape de rééchantillonnage au temps $t = k + 1$ sélectionne n_{k+1} particules de l'échantillon S_k , on obtient ainsi un échantillon de particules approximant la loi cible π_{k+1} , notons que cette étape est similaire à celle utilisée dans le filtre SIR.

L'étape de propagation au temps $t = k + 1$ va nous permettre d'obtenir des particules approximant la loi π_{k+1} , cette étape se fait au travers d'une méthode MCMC. Ce qui est décrit dans l'algorithme 3.3.3 ne fait apparaître qu'une seule itération de cette méthode, en pratique on a besoin de plusieurs itérations ce qui nécessite un certain temps, appelé "burn-in time", avant d'obtenir des particules approximant réellement cette loi.

Pour comprendre cette notion de "burn-in time", rappelons le principe des méthodes MCMC. Leur but est d'échantillonner selon une loi cible, pour cela on construit une chaîne de Markov dont la loi limite est la loi cible, au bout d'un certain temps : "burn-in time" (difficile à déterminer) cette chaîne atteint son état stationnaire ce qui nous permet d'obtenir un échantillon de la loi cible. Il existe plusieurs méthodes MCMC : l'algorithme de Metropolis-Hastings (cf. section 3.3.2 page 31), l'échantillonneur de Gibbs... Dans notre cas, à k et i fixés, la loi cible est π_{k+1} , on utilise l'algorithme de Metropolis-Hastings et on choisit $q_k(\tilde{\theta}_i, \cdot)$ comme loi de proposition, on obtient ainsi après ℓ itérations la chaîne de Markov suivante : $(\theta_k^{i,\ell})_{\ell \geq 1}$.

Revenons à notre algorithme. Les particules rééchantillonnées approximant la loi cible π_{k+1} ce qui donne par l'équation (3.2) un nouvel échantillon de la loi π_{k+1} , le "burn-in time" usuel n'est donc pas nécessaire. En effet, les particules étant "proches" de la loi cible l'état stationnaire sera rapidement atteint. A l'heure actuelle, de nombreux points restent à éclaircir, en effet quelle taille n_k de particules faut-il prendre ? Quel noyau de transition faut-il choisir lors de l'étape de propagation ?

3.3.3 Amélioration de l'échantillonnage séquentiel de Monte Carlo

L'article de Del Moral, A. Doucet et A. Jasra [4] propose également une amélioration du filtrage particulaire. Revenons au cas où l'on cherche à estimer la loi conditionnelle de l'état X_k sachant une réalisation des observations $Y_{1:k} = y_{1:k}$, on note π_k cette loi :

$$\pi_k(dx) = \gamma_k(dx)/Z_k,$$

Hypothèse : Pour simplifier les choses, on suppose dans cette section, que chaque loi $\pi_k(dx)$ admet une densité notée $\pi_k(x)$ par rapport à la mesure de Lebesgue.

Soit Z_k la constante de normalisation et η_k la densité d'importance utilisée lors de l'échantillonnage d'importance. On a alors pour toute fonction φ :

$$\mathbb{E}_{\pi_k}(\varphi) = Z_k^{-1} \int_{\Omega} \varphi(x) w_k(x) \eta_k(x) dx ,$$

où

$$Z_k = \int_{\Omega} w_k(x) \eta_k(x) dx \quad \text{et} \quad w_k(x) = \gamma_k(x)/\eta_k(x) .$$

Supposons qu'au temps $t = k - 1$ nous ayons N particules $\{\xi_{k-1}^{(i)}\} (i = 1 : N)$ échantillonnées selon η_{k-1} , on se propose de propager ces particules selon un noyau de Markov K_k tel que :

$$\eta_k(x') = \int_{\Omega} \eta_{k-1}(x) K_k(x, x') dx .$$

Le problème qui se pose est le calcul de $\eta_k(x)$ dans l'expression des poids. La méthode proposée est d'introduire une technique de variable auxiliaire et des noyaux de Markov "artificiels" notés $L_{k-1}(x, dx')$ de densité $L_{k-1}(x, x')$. On définit alors la loi cible jointe artificielle par

$$\tilde{\pi}_k(x_{1:k}) = \tilde{\gamma}(x_{1:k})/Z_k ,$$

où $\tilde{\gamma}_k(x_{1:k}) = \gamma_k(x_n) \prod_{i=1}^{k-1} L_k(x_{i+1}, x_i)$. Par construction $\tilde{\pi}_k(x_{1:k})$ admet $\pi_k(x_k)$ comme densité marginale.

Passons à la description de l'algorithme qui préserve une complexité de calcul en $\mathcal{O}(N)$. Supposons que l'on possède au temps $t = k - 1$ un échantillon $\{W_{k-1}^i, \xi_{1:k-1}^i\} (i = 1, \dots, N)$ approximant $\tilde{\pi}_{k-1}$, tel que :

$$\begin{aligned} \tilde{\pi}_{k-1}^N(dx_{1:k-1}) &= \sum_{i=1}^N W_{k-1}^i \delta_{\xi_{1:k-1}^i}(dx_{1:k-1}) , \\ W_{k-1}^i &= w_{k-1}(\xi_{1:k-1}^i) / \sum_{i=1}^N w_{k-1}(\xi_{1:k-1}^i) . \end{aligned}$$

où

$$\begin{aligned} w_k(x_{1:k}) &= \tilde{\gamma}_k(x_{1:k})/\eta_k(x_{1:k}) \\ &= w_{k-1}(x_{1:k-1}) \tilde{w}(x_{k-1}, x_k) , \end{aligned}$$

et

$$\tilde{w}(x_{k-1}, x_k) = \frac{\gamma_k(x_k) L_{k-1}(x_k, x_{k-1})}{\gamma_{k-1}(x_{k-1}) K_{k-1}(x_{k-1}, x_k)}.$$

Algorithme 3.3.4 (*Échantillonnage séquentiel de Monte Carlo*)

```

 $\xi_1^{1:N} \sim \eta_1$ 
 $W_1^i \leftarrow \gamma_1(\xi_1^i) / \eta_1(\xi_1^i)$ 
 $k \leftarrow k + 1$ 
while  $k \leq T$  do
  if  $N_{eff}^2 < \text{taux}$  then %pour un seuil déterminé au préalable
     $\xi_k^{1:N} \leftarrow \text{resample}(W_k^{1:N}, \xi_k^{1:N})$ 
     $W_k^{1:N} \leftarrow 1/N$ 
  end
   $k \leftarrow k + 1$ 
  if  $k = T + 1$  then %on arrête l'algorithme
    sortie  $(W_T^{1:N}, \xi_{1:T}^{1:N})$ 
  else
     $\xi_k^i \sim K_k(\xi_{k-1}^i, \cdot)$  pour  $i = 1 : N$  %propagation
     $\tilde{W}_k^i \leftarrow W_{k-1}^i \frac{\gamma_k(\xi_k^i) L_{k-1}(\xi_k^i, \xi_{k-1}^i)}{\gamma_{k-1}(\xi_{k-1}^i) K_{k-1}(\xi_{k-1}^i, \xi_k^i)}$  pour  $i = 1 : N$  %mise à jour
  des poids
     $W_k^i \leftarrow \tilde{W}_k^i / \sum_{j=1}^N \tilde{W}_k^j$  pour  $i = 1 : N$  %normalisation
  end
end

```

Au temps k on obtient, après la propagation, un système de particules $\{W_k^{(i)}, \xi_{1:k}^{(i)}\}$ approximant $\tilde{\pi}_k(x_{1:k})$. La loi $\pi_k(x_k)$ étant une loi marginale de $\tilde{\pi}_k(x_{1:k})$, on a que :

$$\pi_k^N(dx) = \sum_{i=1}^N W_k^{(i)} \delta_{\xi_k^{(i)}}(dx).$$

La proposition suivante propose une suite de noyaux intermédiaires optimale, qui n'est pas utilisable dans la pratique.

Proposition 3.3.5 *La suite de noyaux intermédiaires optimale est*

$$L_{k-1}^{opt}(x_k, x_{k-1}) = \frac{\eta_{k-1}(x_{k-1}) K_k(x_{k-1}, x_k)}{\eta_k(x_k)},$$

et dans ce cas

$$w_k(x_{1:k}) = \gamma_k(x_k) / \eta_k(x_k).$$

Plusieurs approximations de cette suite existent, retenons les deux principales :

- La première approche est de remplacer η_{k-1} par π_{k-1} , ce qui nous donne :

$$L_{k-1}(x_k, x_{k-1}) = \frac{\pi_{k-1}(x_{k-1}) K_k(x_{k-1}, x_k)}{\pi_k(x_k)}, \quad (3.3)$$

on a ainsi

$$\tilde{w}(x_{k-1}, x_k) = \frac{\gamma_k(x_k)}{\int_{\mathbb{R}^d} \gamma_{k-1}(x_{k-1}) K_{k-1}(x_{k-1}, x_k) dx_{k-1}}.$$

- Dans le cas où K_k est un noyau d'une méthode MCMC de loi stationnaire π_k , la deuxième approche consiste à choisir :

$$L_{k-1}(x_k, x_{k-1}) = \frac{\pi_k(x_{k-1}) K_k(x_{k-1}, x_k)}{\pi_k(x_k)}, \quad (3.4)$$

et dans ce cas :

$$\tilde{w}(x_{k-1}, x_k) = \frac{\gamma_k(x_k)}{\gamma_{k-1}(x_{k-1})}.$$

Le choix d'une suite $\{L_k\}$ aussi proche que possible de $\{L_k^{opt}\}$ est primordial pour que notre méthode soit efficace. Dans le cas où K_k est un noyau d'une méthode MCMC de loi stationnaire π_k , on peut mettre cette méthode en parallèle avec des techniques connues, notamment le resample-move algorithm (Gilks et Berzuini, 2001) et l'AIS (Neal, 2001). En effet, ces différentes techniques utilisent implicitement la suite de noyaux intermédiaires définie dans l'expression (3.4). Pour ces algorithmes le poids associé est de la forme suivante :

$$\tilde{w}_k(\xi_{k-1}^{(i)}, \xi_k^{(i)}) \propto \pi_k(\xi_{k-1}^{(i)}) / \pi_{k-1}(\xi_{k-1}^{(i)}). \quad (3.5)$$

Le problème avec l'expression (3.5) est que les poids sont indépendants de $\{\xi_k^{(i)}\}$ où $\xi_k^{(i)} \sim K_k(\xi_{k-1}^{(i)}, \cdot)$. En particulier, la variance de cette expression sera grande si la différence entre π_{k-1} et π_k est importante et ceci même si le

noyau K_k est “bon”. Par contre, le choix de l’expression (3.3) permet d’obtenir des poids tels que :

$$\tilde{w}_k(\xi_{k-1}^{(i)}, \xi_k^{(i)}) \propto \pi_k(\xi_k^{(i)}) / \pi_{k-1} K_k(\xi_k^{(i)}). \quad (3.6)$$

ce qui est plus intuitif. Un simple critère de Rao-Blackwell permet de montrer que la variance de (3.6) est toujours plus petite que celle de (3.5).

En ce qui concerne les résultats, l’article propose une comparaison avec l’AIS. On constate un gain en ce qui concerne l’estimation des valeurs autres que les constantes normalisées.

Pour résumer, cette nouvelle méthode est “potentiellement puissante” sous certaines conditions, en effet elle dépend du choix du noyau K_k , de la loi cible π_k et du temps qui doit être modéré.

Chapitre 4

Applications

4.1 Application à la pêche

La modélisation bayésienne, flexible, est adaptée au traitement de modèles complexes rencontrés dans le domaine de l'écologie, nous illustrerons cet aspect par un exemple dans le domaine de l'halieutique.

4.1.1 Données et méthodes

En halieutique, les modèles bayésiens se révèlent particulièrement utiles compte tenu de l'incertitude liée à la complexité des écosystèmes marins et aux problèmes de collecte de données. Nous étudierons ici l'état du stock de poulpe à l'échelle de la ZEE mauritanienne, les données récoltées entre 1971 et 2005 proviennent de la Délégation à la Surveillance Pêche et au Contrôle en Mer (DSPCM). Elles ont été transmises par Étienne Rivot dans le cadre d'une collaboration avec l'Agrocampus de Rennes. Elles sont données en annexe (cf. B), elles représentent :

- Les indices d'abondance (notés IA), ces indices reflètent l'évolution de la taille de la population.
- Les captures pour chaque année.

Les techniques d'estimation des stocks de pêche visent à évaluer l'impact de différents scénarios d'exploitation et de capture sur l'évolution de l'abondance d'un stock de poissons. Cette abondance s'exprime en poids total de poissons vulnérables à la pêche (biomasse). L'accroissement de cette biomasse intègre l'augmentation du poids moyen de chaque individu ainsi que le recrutement, c'est à dire l'arrivée chaque année de nouvelles générations de poissons. À

l'inverse, cette biomasse décroît du fait de la capture (mortalité par pêche) ou d'autres causes (mortalité naturelle : prédation, maladies, etc.). Le modèle que nous utiliserons, pour modéliser cette biomasse, est le modèle exponentiel de Fox (1970), pour plus de détails voir [13]. L'équation d'état s'écrit :

$$B_{t+1} = [B_t + r B_t (1 - \frac{\log(B_t)}{\log(K)})] \exp(\varepsilon_t) - C_t,$$

où B_t représente la biomasse au début de l'année t , r est le taux intrinsèque de croissance, K est la capacité biotique, C_t représente la capture de poulpe à l'année t et ε_t suit une loi de distribution gaussienne de moyenne 0 et de variance σ pour tout t .

L'équation d'observation permet de relier les indices d'abondances observés à la biomasse :

$$I_t = q B_t \exp(\sigma_t),$$

où I_t représente l'indice d'abondance à l'année t et q la capturabilité. σ_t suit une loi de distribution gaussienne de moyenne 0 et de variance τ pour tout t . Pour simplifier les notations on passe à la forme logarithmique des observations, on obtient donc :

$$Y_t = \log(q) + \log(B_t) + \sigma_t.$$

On a donc un modèle de Markov caché avec paramètres, notre but sera d'approximer la loi :

$$\pi_k \stackrel{\text{d\u00e9f}}{=} \text{loi}(X_k, r, K, q | Y_{1:k} = y_{1:k}) \quad \forall k \geq 1.$$

L'état de stock de poulpe en Mauritanie a été évalué à plusieurs reprises par les méthodes de production. En particulier, le groupe de travail COPACE 2004 (sous presse) et plus récemment le GT Démersal IMROP-RIVO 2006 (ouldMahmoud et al., 2006) ont permis d'évaluer l'état du stock de poulpe (*Octopus vulgaris*) et de la pêche à partir de modèles globaux. Les différentes études menées jusqu'à présent, ont montré que la biomasse actuelle est largement en-dessous de celle qui produit le rendement maximal soutenable et la mortalité due à la pêche est très élevée par rapport à celle nécessaire pour capturer toute la production naturelle du stock. Au vue de ces résultats alarmants, l'étude de ce stock et les améliorations possibles constituent des enjeux économiques importants.

Nous utilisons le filtre bootstrap et l'ASIR pour évaluer le stock de poulpe et les paramètres. Puis nous comparons nos résultats à ceux obtenus par le groupe de travail d'Étienne Rivot avec une méthode MCMC : l'algorithme de Metropolis-Hasting.

4.1.2 Résultats

Avant la première année $t = 1$, les captures sont supposées nulles. La biomasse initiale (année 1971) est supposée égale à 90% de la capacité biotique et on considère les paramètres K , r et q fixes, les valeurs données sont celles obtenues a posteriori par les groupes de travaux précédents. Cette première approche nous permet d'avoir une idée de l'évolution de la biomasse au cours de ces 26 dernières années.

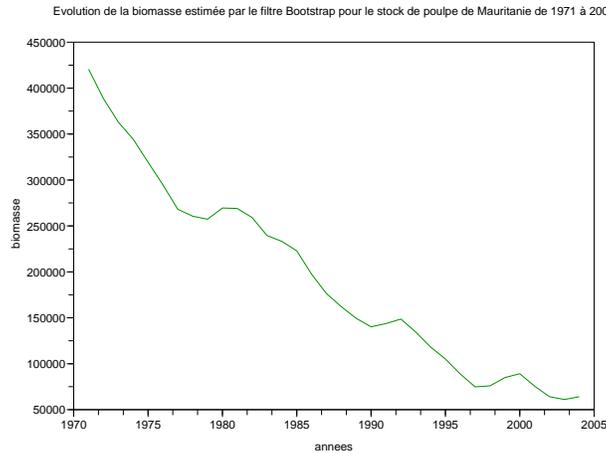


FIG. 4.1 – Estimation de la biomasse par l'ASIR avec 5000 particules, $K = 460000$, $r = 2$, $q = 1/120000$ entre 1971 et 2005.

La figure 4.1 représente l'estimation de la biomasse par l'ASIR à paramètres fixés avec 5000 particules, $\tau = 0.1$ et $\sigma = 0.001$ entre 1971 et 2005. On constate la baisse continue du stock de poulpe. Passons maintenant au cas où l'on met des lois a priori sur nos paramètres. Les valeurs des paramètres r et K estimées par les modèles globaux dans les précédents groupes de travail permettent de définir une loi de distribution a priori informative pour r et

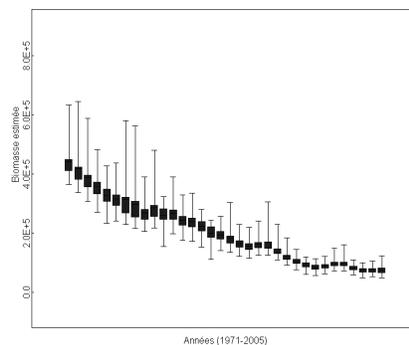


FIG. 4.2 – Éstimation de l'évolution de la biomasse par une méthode MCMC.

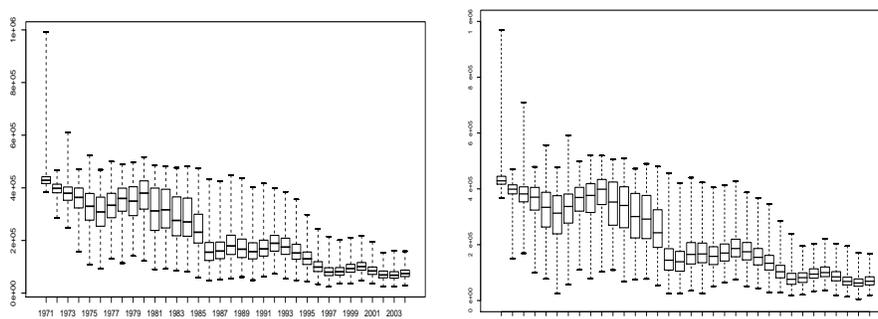


FIG. 4.3 – Éstimation de la biomasse entre 1976 et 2004 : ASIR et 5000 particules (gauche) / filtre bootstrap et 5000 particules (droite).

de prendre $K = 460000$. On considère une loi de distribution a priori non informative pour le paramètre q :

$$\begin{aligned} \log(r) &\sim \mathcal{N}(1.1, 1.5), \\ q^{-1} &\sim \mathcal{U}_{[10000, 1000000]}. \end{aligned}$$

Pour appliquer le filtre bootstrap et l'ASIR, nous allons traiter nos paramètres comme des variables d'état, la loi d'évolution de nos paramètres est telle que :

$$r_{t+1} = r_t, \text{ et } q_{t+1} = q_t.$$

La figure 4.2 montre l'estimation de la biomasse obtenue par une méthode

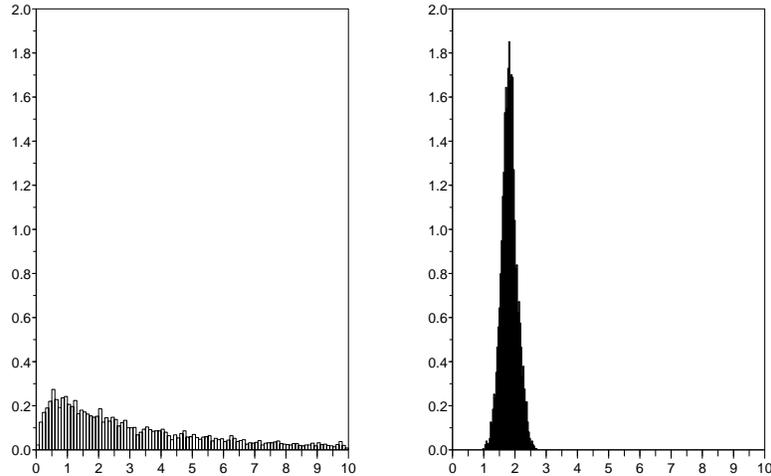


FIG. 4.4 – Lois de distribution a priori (à gauche) et a posteriori (à droite) pour le taux intrinsèque de croissance r obtenu par l'ASIR avec 5000 particules.

MCMC. La figure 4.3 présente cette même estimation effectuée par l'ASIR avec 5000 particules (à gauche) et par le filtre bootstrap avec 10000 particules (à droite). On prend : $\tau = 0.1$ et $\sigma = 0.01$. L'estimation obtenue par les filtres particuliers est très ressemblante à celle obtenue par une méthode MCMC. En effet, la tendance est la même : la biomasse du stock aurait diminué de plus de 80% entre 1971 et 2005. On note toutefois que les filtres particuliers présentent un peu plus de variance, ceci est cohérent avec le fait que les filtres particuliers donnent une approximation de $\text{loi}(X_t|Y_{1:t})$ alors que la méthode MCMC donne une approximation de $\text{loi}(X_t|Y_{1:T})$. Il est possible de remédier à cette différence technique en utilisant le lisseur particulaire. Cependant nous n'avons pas considéré cette méthode car elle n'est pas aussi aboutie que le filtre particulaire. Son implémentation pose de nombreux problèmes, notamment la dégénérescence très rapide des poids et ce malgré une étape de rééchantillonnage.

La figure 4.3 permet également de constater que l'ASIR présente moins de variance que le filtre Bootstrap. Dans la suite, on expose uniquement les résultats obtenus par l'ASIR.

L'information contenue dans la série de données a permis de “mettre à jour” la loi de distribution a posteriori du paramètre r , ce qui est illustré par la figure 4.4.

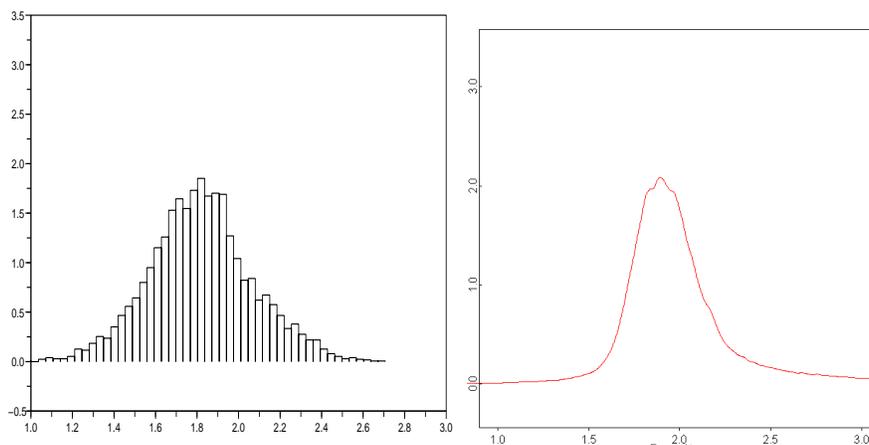


FIG. 4.5 – Comparaison filtrage particulaire (gauche) et MCMC (droite) pour la loi a posteriori de r .

Dans la figure 4.5, nous comparons l’ASIR (avec 5000 particules) et la méthode MCMC pour la loi a posteriori du paramètre r . Une nouvelle fois, les résultats sont comparables. Revenons à l’algorithme. Le filtre particulaire “classique” n’effectue pas de mise à jour des paramètres, en effet la loi de transition les laisse constants, l’espace des paramètres n’est exploré qu’une seule fois (lors de l’initialisation) ce qui conduit à la divergence de l’algorithme, pour contourner ce problème nous avons utilisé la méthode de régularisation avec noyau de convolution. Cette méthode peut se voir comme l’ajout d’un bruit artificiel. On l’explique pour le paramètre r . Au temps t on possède, après l’étape de rééchantillonnage, un échantillon $\{\tilde{r}^i\}(i = 1 : N)$ de la loi $\mathbb{P}(dr|y_{1:t})$, la méthode du noyau de convolution consiste à utiliser l’approximation suivante :

$$p(r^i|y_{1:t}) \simeq \frac{1}{N} \sum_{i=1}^N K_{h_n}(\tilde{r}^i - r^i),$$

où $h_n = A_r(N)^{-1/5} \sigma_r$ et K est le noyau gaussien, et à simuler N particules tel que :

$$r^i \sim K_{h_n}(\tilde{r}^i - r^i).$$

Pour plus de détails voir [3]. Ainsi pour la simulation précédente nous avons appliqué cette méthode pour les paramètres r, q en prenant $A_r = 1, A_q = 0.03$.

Pour résumer, on constate que le filtrage particulière et les méthodes MCMC convergent vers les mêmes résultats, ce qui nous amène à penser que les filtres particulières ont utilisé un nombre de particules suffisant et que la méthode MCMC a atteint son état stationnaire.

4.2 Application à la foresterie

4.2.1 Introduction

La modélisation forestière a pour objectif de faire progresser les modes de gestion du patrimoine pour préserver et améliorer ses ressources, mais aussi pour assurer un bon fonctionnement de son écosystème. Nous conseillons la lecture du livre de A. Franc, S. Gourlet-fleuri et N. Picard [1] pour une première approche de la modélisation des forêts hétérogènes. On entend par forêt hétérogène une forêt qui présente dans l'espace un mélange de structures, d'âges, de tailles ou de toute autre variable qui permette de la décrire localement. On peut prendre par exemple le mélange d'espèces ; c'est le cas de la forêt guyanaise où le nombre d'espèces est en moyenne de 150 à l'hectare.

4.2.2 Données et méthodes

Nos données sont les arbres des parcelles expérimentales du dispositif de Paracou en Guyane française. Pour être précis nous disposons des parcelles témoins 1, 6, 11 ; chaque parcelle est divisée en 4 carrés, nous prendrons le diamètre de l'arbre comme variable. Le modèle utilisé est un modèle matriciel : le modèle d'Usher. Les modèles matriciels ont été introduits par Bernardelli (1941) en écologie et par Lewis (1942) et Leslie (1945) dans le contexte de la démographie humaine. Ils ont été rapidement étendus à la gestion des ressources renouvelables, notamment forestière par Usher (1966). On supposera que les évolutions des individus dépendent de la densité courante de la population, notre modèle est donc un modèle densité-dépendant, ces modèles rendent compte des phénomènes biologiques et environnementaux de régulation dans l'évolution d'une population, pour plus de détails nous conseillons la thèse de M. Zetlaoui (voir [15]).

Nous allons répartir notre population en 4 classes de diamètre (10 – 20 cm, 20 – 30 cm, 30 – 40 cm, plus de 40 cm). L'évolution de la population est décrite par le vecteur d'effectif, noté $N_t = (N_t^i)_{i=1, \dots, 4}$ et définie entre le temps t et $t + 1$ par la relation :

$$N_{t+1} = U_t N_t,$$

la densité-dépendance signifie que la matrice U_t (ou de façon équivalente, les taux de transition p_i , q_i et f) dépendent du vecteur N_t , c'est-à-dire :

$$U_t = U(N_t).$$

Notre matrice U_t est de la forme suivante :

$$U_t = \begin{pmatrix} p_1^t + f^t & f^t & f^t & f^t \\ q_2^t & p_2^t & 0 & 0 \\ 0 & q_3^t & p_3^t & 0 \\ 0 & 0 & q_4^t & p_4^t \end{pmatrix}$$

où le paramètre f^t s'interprètent comme la fertilité des individus, les paramètres p_i^t sont les taux de survis sans croissance entre le temps t et $t + 1$, les paramètres q_i^t sont les taux de survie avec croissance entre le temps t et $t + 1$.

L'objectif est d'estimer les coefficients de la matrice U_t pour tout $t \geq 1$. Dans notre application nous disposons des effectifs des années 1984 – 1995 et 1997, 1999, 2001, 2003. L'évolution du vecteur N_t se décrit donc de la façon suivante :

$$N_{t+1} = U(\theta, N_t) N_t \tag{4.1}$$

où θ est un paramètre inconnu. Dans le cas linéaire $U(\theta, N_t) = U(\theta)$.

Modélisation probabiliste

Nous allons décrire l'évolution du vecteur N_t sous forme markovienne. Le comportement moyen de cette chaîne de Markov sera donné par 4.1. Le modèle est donc $(N_{1:T}, \theta)$ et sa loi est donnée par :

$$\text{loi}(N_{1:T}, \theta) = \text{loi}(\theta) \text{loi}(N_{1:T}|\theta)$$

avec :

$$\begin{aligned}\text{loi}(N_{1:T}|\theta) &= \text{loi}(N_1|\theta) \prod_{t=1}^{T-1} \text{loi}(N_{t+1}|N_t, \theta) \\ &= \text{loi}(N_1|\theta) \prod_{t=1}^{T-1} \prod_{i=1}^4 \text{loi}(N_{t+1}^i|N_t, \theta)\end{aligned}$$

On suppose donc que, conditionnellement à θ , N_t est une chaîne de Markov et que son noyau de transition est le produit des 4 composantes (pour $i = 1$ à 4). Nous décrivons maintenant le noyau de transition de N_t (à θ fixé).

Noyau de transition

L'évolution du vecteur effectif entre le temps t et $t + 1$ est donnée par la relation suivante :

$$N_{t+1}^i = F_{ii}(t) + F_{(i-1)i}(t),$$

c'est-à-dire que le nombre d'arbres au temps $t + 1$ dans la classe i est égal au nombre d'arbres qui étaient dans la classe i et qui y sont restés entre le temps t et $t + 1$ ($F_{ii}(t)$), plus le nombre d'arbres qui étaient dans la classe $i - 1$ et qui sont passés dans la classe i entre le temps t et $t + 1$ ($F_{(i-1)i}(t)$) (avec pour convention $F_{01}(t) = r(t)$ où $r(t)$ représente le nombre de nouveaux individus).

Notations :

$$\begin{aligned}p_i^t &= p_i(N_t) \\ q_{(i+1)}^t &= q_{i+1}(N_t),\end{aligned}$$

Revenons à notre vraisemblance, pour tout $m = (m_1, m_2, m_3, m_4)$ et $n = (n_1, n_2, n_3, n_4)$ de \mathbb{N}^4 , on a :

$$\mathbb{P}(N_{t+1}^i = n_i | N_t = m, \theta) = \mathbb{P}(F_{ii}(t) + F_{(i-1)i}(t) = n_i | N_t = m, \theta). \quad (4.2)$$

$F_{ii}(t)$ et $F_{(i-1)i}(t)$ sont indépendants conditionnellement aux paramètres et à l'état, de plus :

$$\begin{aligned}F_{ii}(t) | N_t = m, \theta &\sim \mathcal{B}(m_i, p_i(m)) \\ F_{(i-1)i}(t) | N_t = m, \theta &\sim \mathcal{B}(m_{i-1}, q_i(m)) \\ F_{01}(t) | N_t = m, \theta &\sim \mathcal{P}\left(f \sum_{i=1}^4 m_i\right).\end{aligned}$$

On a la convolution de deux binomiales, ce qui donne dans l'égalité 4.2, $\forall i \geq 2$:

$$\begin{aligned} \mathbb{P}(N_{t+1}^i = n_i | N_t = m, \theta) &= \mathbb{P}(F_{ii}(t) + F_{(i-1)i}(t) = n_i | N_t = m, \theta) \\ &= \sum_k \mathbb{P}(F_{ii}(t) = k | N_t = m, \theta) \mathbb{P}(F_{(i-1)i}(t) = n_i - k | N_t = m, \theta) \\ &= \sum_k C_{m_i}^k (p_i(m))^k (1 - p_i(m))^{m_i - k} C_{m_{i-1}}^{n_i - k} (q_i(m))^{n_i - k} (1 - q_i(m))^{m_{i-1} - n_i + k} \end{aligned}$$

où $k \in \{n_i - \min(m_{i-1}, n_i), \min(m_{i-1}, n_i)\}$.

Pour $i = 1$, on a :

$$\begin{aligned} \mathbb{P}(N_{t+1}^1 = n_1 | N_t = m, \theta) &= \mathbb{P}(F_{11}(t) + F_{01}(t) = n_1 | N_t = m, \theta) \\ &= \sum_k \mathbb{P}(F_{11}(t) = k | N_t = m, \theta) \mathbb{P}(F_{01}(t) = n_1 - k | N_t = m, \theta) \\ &= \sum_{k=0}^{\min(m_1, n_1)} C_{m_1}^k (p_1(m))^k (1 - p_1(m))^{m_1 - k} \frac{(\sum_{j=1}^4 f m_j)^{n_1 - k}}{n_1 - k} \exp(- (f \sum_{j=1}^4 m_j)). \end{aligned}$$

On obtient $\forall t \geq 1$:

$$\begin{aligned} \mathbb{P}(N_{t+1} = n | N_t = m, \theta) &= \prod_{i=1}^4 \mathbb{P}(N_{t+1}^i = n_i | N_t = m, \theta) \\ &= \prod_{i=2}^4 \left[\sum_{k=0}^{\min(m_{i-1}, n_i)} C_{m_i}^k (p_i(m))^k (1 - p_i(m))^{m_i - k} C_{m_{i-1}}^{n_i - k} (q_i(m))^{n_i - k} (1 - q_i(m))^{m_{i-1} - n_i + k} \right] \\ &\quad \sum_{k=0}^{\min(m_1, n_1)} C_{m_1}^k (p_1(m))^k (1 - p_1(m))^{m_1 - k} \frac{(\sum_{j=1}^4 f m_j)^{n_1 - k}}{n_1 - k} \exp(- (f \sum_{j=1}^4 m_j)). \end{aligned}$$

Approximation gaussienne

Les expressions précédentes sont peu pratiques. Nous allons donc utiliser une approximation gaussienne. Rappelons que les lois binomiales et de Pois-

son peuvent être approchées, sous certaines conditions, par une loi normale :

$$\begin{aligned}\mathcal{B}(n, p) &\cong \mathcal{N}(np, np(1-p)) \\ \mathbb{P}(\lambda) &\cong \mathcal{N}(\lambda, \lambda).\end{aligned}$$

Ce qui donne :

$$\begin{aligned}F_{ii}(t)|N_t = m, \theta &\sim \mathcal{N}(m_i p_i(m), m_i p_i(m)(1-p_i(m))), \\ F_{(i-1)i}(t)|N_t = m, \theta &\sim \mathcal{N}(m_{i-1} q_i(m), m_{i-1} q_i(m)(1-q_i(m))), \\ F_{01}(t)|N_t = m, \theta &\sim \mathcal{N}(f^t \sum_{j=1}^4 m_j, f^t \sum_{j=1}^4 m_j).\end{aligned}$$

Ainsi, $\forall i \geq 2$:

$$N_{t+1}^i | N_t = m, \theta \sim \mathcal{N}(m_i p_i(m) + m_{i-1} q_i(m), m_i p_i(m)(1-p_i(m)) + m_{i-1} q_i(m)(1-q_i(m)))$$

et

$$N_{t+1}^1 | N_t = m, \theta \sim \mathcal{N}(m_1 p_1(m) + f^t \sum_{j=1}^4 m_j, m_1 p_1(m)(1-p_1(m)) + f^t \sum_{j=1}^4 m_j)$$

Donc

$$N_{t+1} | N_t = m, \theta \sim \mathcal{N}(\mu, \Sigma),$$

$$\text{où } \mu = \begin{pmatrix} m_1 p_1(m) + f^t \\ m_2 p_2(m) + m_1 q_2(m) \\ m_3 p_3(m) + m_2 q_3(m) \\ m_4 p_4(m) + m_3 q_4(m) \end{pmatrix} \text{ et } \Sigma = \begin{pmatrix} m_1 p_1(m)(1-p_1(m)) + f^t \sum_{j=1}^4 m_j & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix}$$

avec $\sigma_i^2 = m_i p_i(m)(1-p_i(m)) + m_{i-1} q_i(m)(1-q_i(m))$.

La suite de notre travail consiste à appliquer le filtre bootstrap pour estimer les taux.

4.2.3 Mise en œuvre et résultats

Modèle de Usher linéaire

Dans un premier temps, on considère nos paramètres constants $\theta = (p_1, p_2, p_3, p_4, q_2, q_3, q_4, f)$.

Initialisation : Pour $i = 1, \dots, N$

$$\begin{pmatrix} p_1^i \\ p_2^i \\ p_3^i \\ p_4^i \end{pmatrix} \sim \text{Beta}(7, 3) , \quad \begin{pmatrix} q_2^i \\ q_3^i \\ q_4^i \end{pmatrix} \sim \mathcal{U}\left(1 - \begin{pmatrix} p_1^i \\ p_2^i \\ p_3^i \end{pmatrix}\right) \text{ et } f^i \sim \mathcal{U}_{[0,1]} .$$

Au temps t : (i) mise à jour des poids :

$$w_t^i \propto p(N_{t+1}|N_t, \theta^i) \text{ pour } i = 1, \dots, N$$

(ii) sélection des particules :

$$\tilde{\theta}^{1:N} \leftarrow \text{multinomiale}(w_t^{1:N}, \theta^{1:N})$$

(iii) mise à jour des paramètres :

$$\theta^i \sim K_{h_n}(\tilde{\theta}^i - \theta^i)$$

Nous allons «tester» le pouvoir de prédiction de cette méthode. Expliquons notre démarche : dans un premier temps on utilise les 11 premières années pour estimer les paramètres. On a donc N particules pour chaque paramètre $(\theta^1, \dots, \theta^N)$. Pour chaque particule on applique le modèle de Usher linéaire, on obtient les lois a posteriori des effectifs de chaque classe. On réalise cette opération pour les années 1995 à 2003.

Voici les résultats. Chaque figure représente la loi a posteriori des effectifs des 4 classes obtenue par le filtre bootstrap et 10000 particules, la croix est la vraie valeur. Ces différents graphiques montrent un pouvoir de prédiction assez faible, en effet les vraies valeurs ne sont pas systématiquement autour du mode des lois a posteriori. Ce problème va se résoudre en passant à un modèle de Usher densité-dépendant. Ce qui nous amène à penser que la croissance dépend de l'effectif.

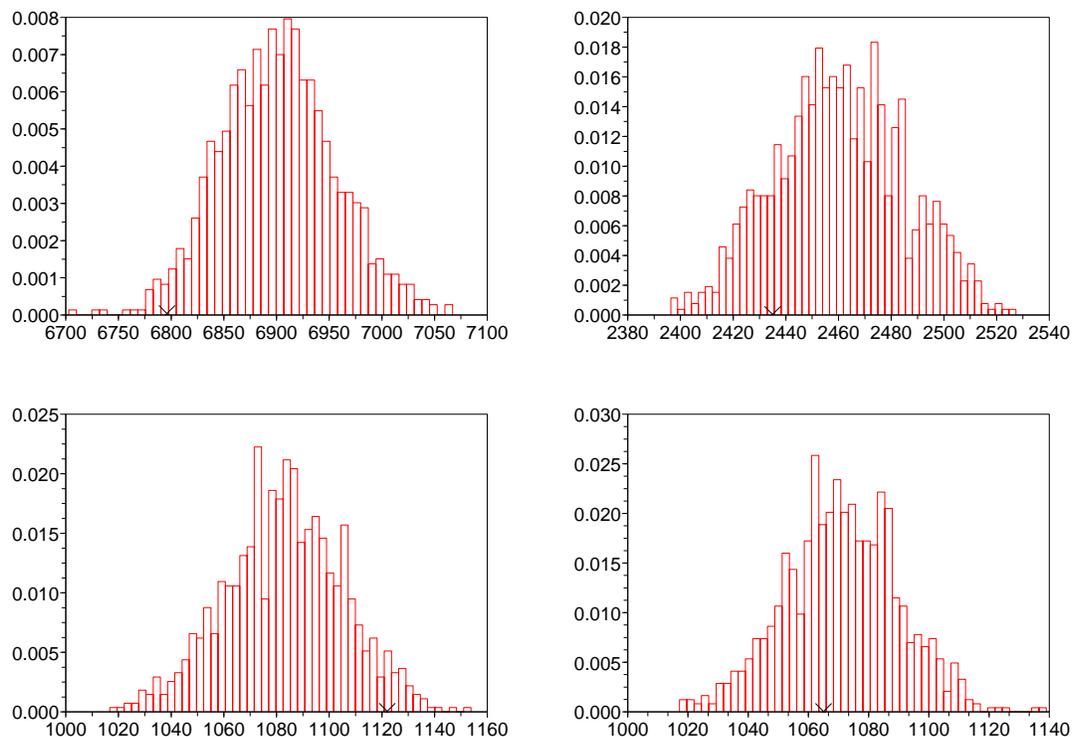


FIG. 4.6 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 1995$.

Modèle de Usher densité-dépendant

On considère maintenant que nos paramètres dépendent de l'effectif, on pose :

$$\text{logit}(p_i^t) = \alpha_i \sum_{j \geq i} N_t^j + \beta_i, \forall i \in \{1, \dots, 4\}$$

$$\text{logit}(q_i^t) = \gamma_i \sum_{j \geq i-1} N_t^j + \delta_i, \forall i \in \{2, \dots, 4\}$$

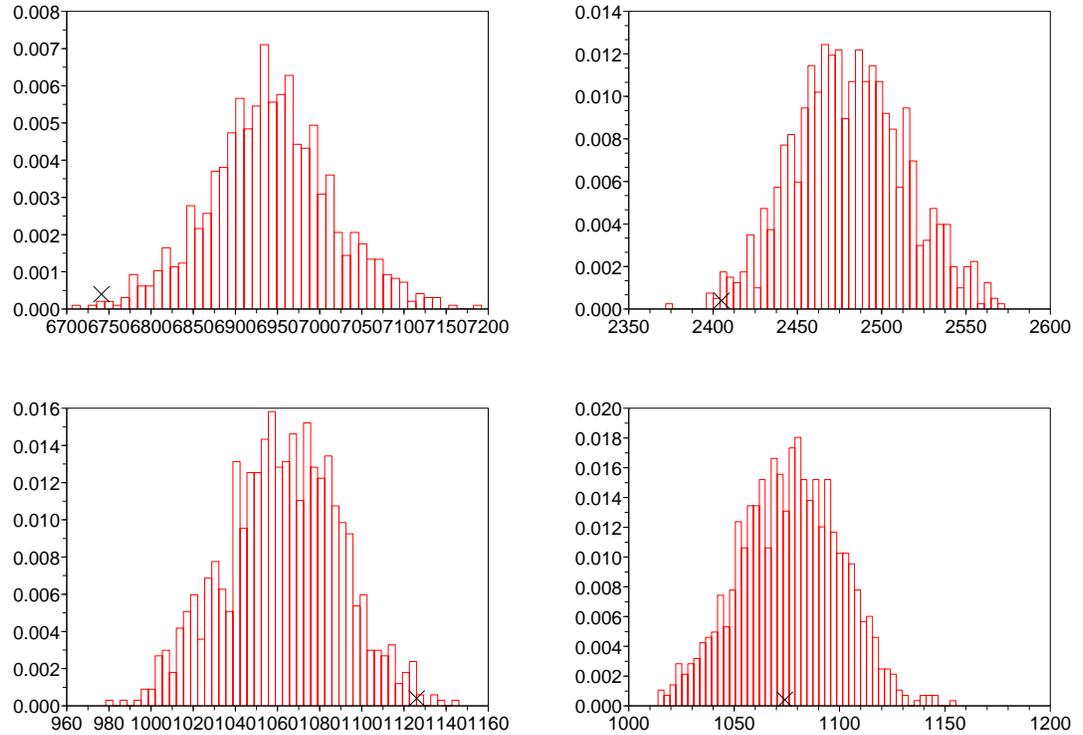


FIG. 4.7 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 1997$.

Ce qui donne :

$$p_i^t = \frac{\exp(\text{logit}(p_i^t))}{1 + \exp(\text{logit}(p_i^t))}, \forall i \in \{1, \dots, 4\}$$

$$q_i^t = \frac{\exp(\text{logit}(q_i^t))}{1 + \exp(\text{logit}(q_i^t))}, \forall i \in \{2, \dots, 4\}$$

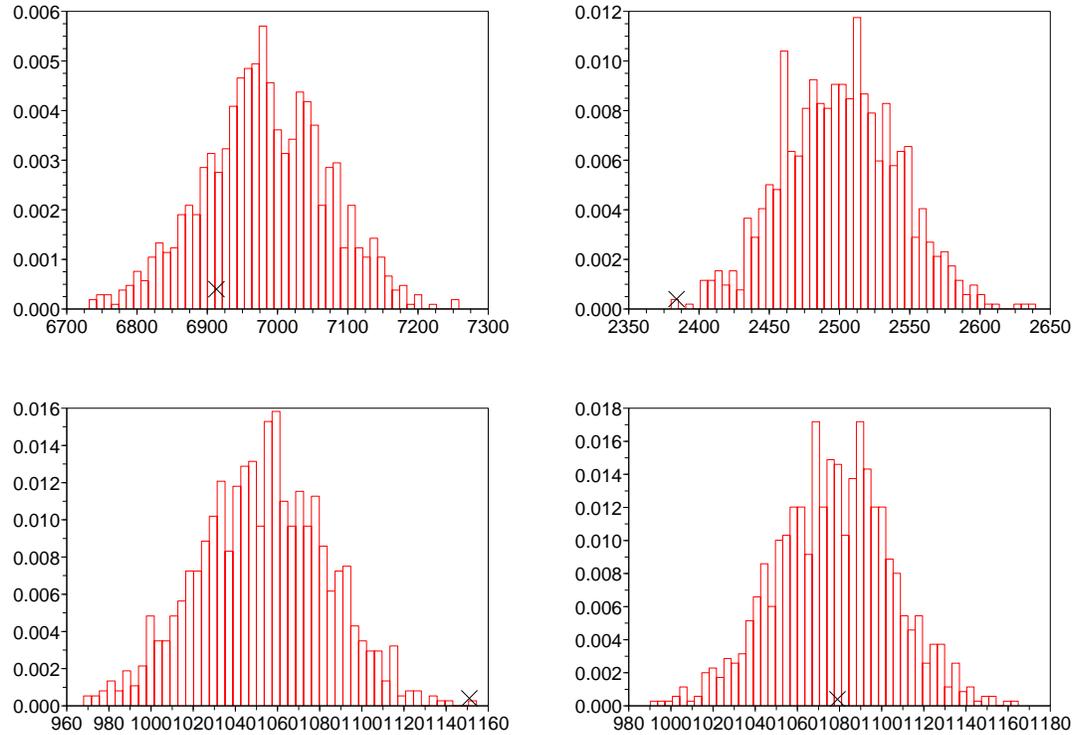


FIG. 4.8 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 1999$.

Notre but est alors d'estimer :

$$\theta = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_2, \gamma_3, \gamma_4, \delta_2, \delta_3, \delta_4, f).$$

Le principe reste le même, on met des particules sur toutes les composantes de θ .

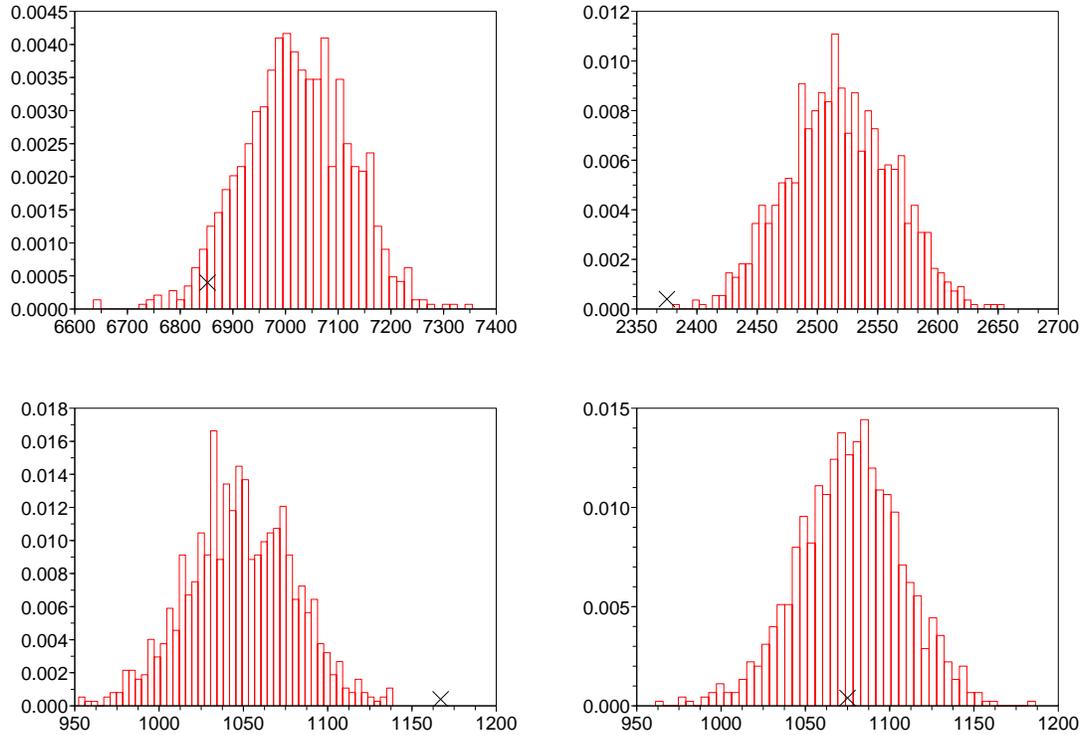


FIG. 4.9 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 2001$.

Initialisation :

$$\theta^{1:N} \sim \mathcal{N}(0, (0.001)^2)$$

Au temps t : (i) mise à jour des poids :

$$w_t^i \propto p(N_{t+1}|N_t, \theta^i) \text{ pour } i = 1, \dots, N$$

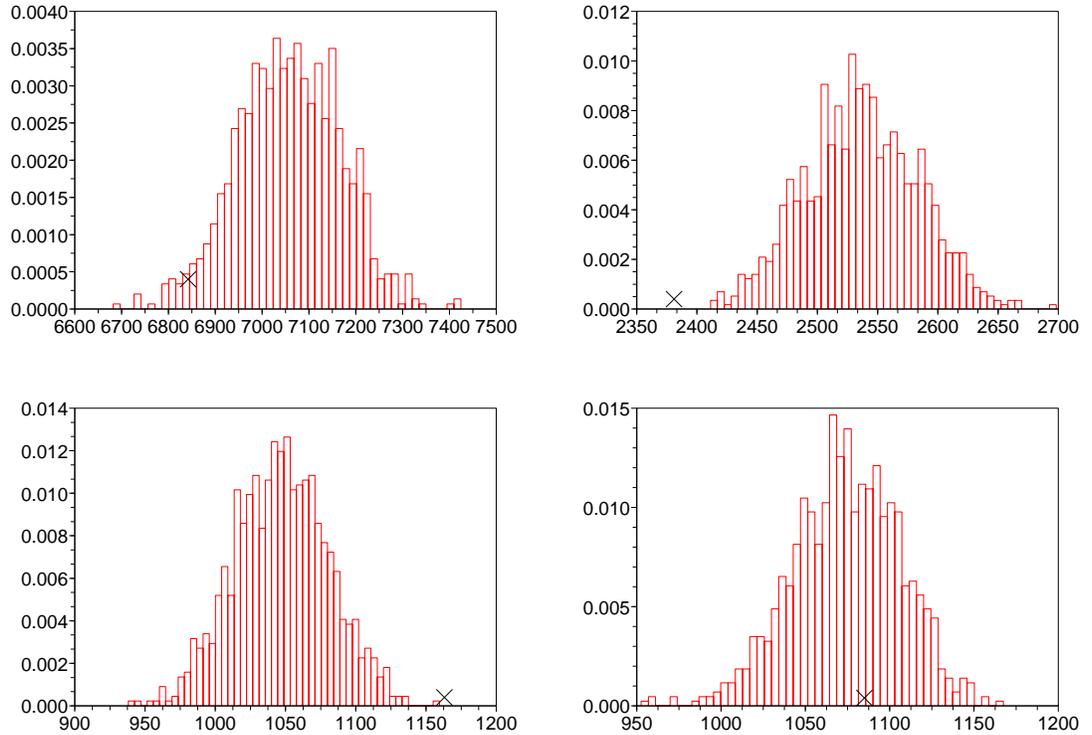


FIG. 4.10 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 2003$.

(ii) sélection des particules :

$$\tilde{\theta}^{1:N} \leftarrow \text{multinomiale}(w_t^{1:N}, \theta^{1:N})$$

(iii) mise à jour des paramètres :

$$\theta^i \sim K_{h_n}(\tilde{\theta}^i - \theta^i)$$

Le modèle de Usher densité-dépendant est très prédictif. En effet, la vraie valeur se situe dans la majorité des cas aux alentours du mode de la loi a posteriori. Ce qui nous conforte dans l'idée que la croissance des arbres dépend de la densité courante de la population. Les 5 graphiques qui suivent représentent la loi a posteriori obtenue par le filtre bootstrap (10000 particules) et la vraie valeur.

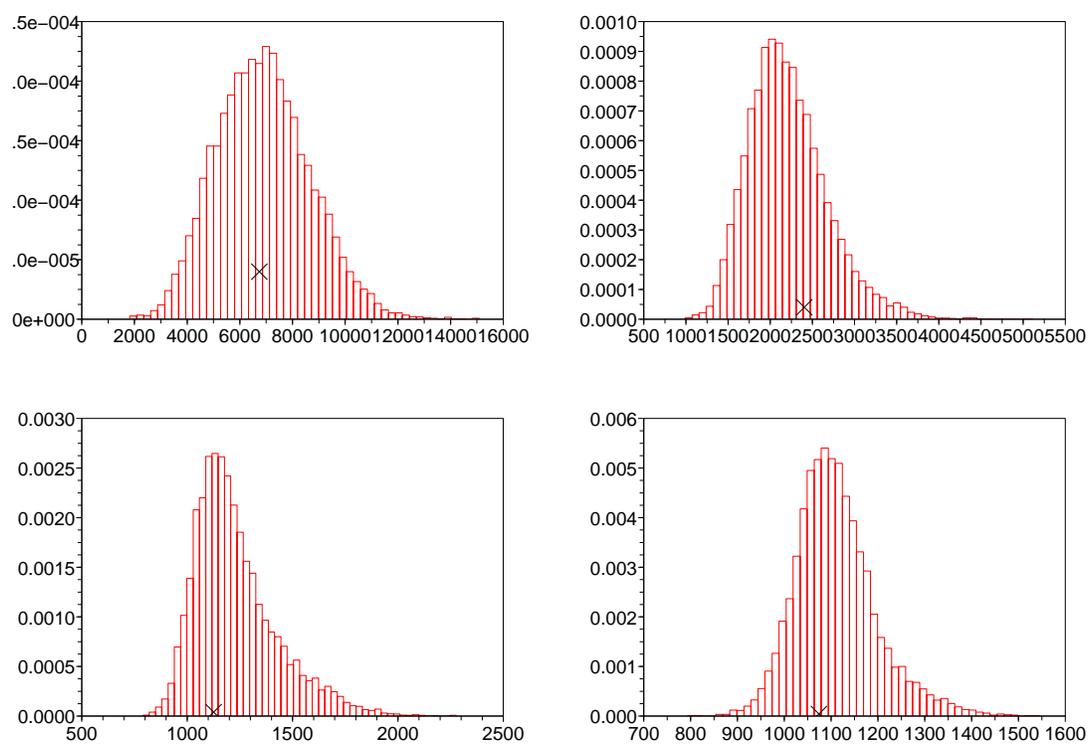


FIG. 4.11 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 1995$ (modèle densité-dépendant).

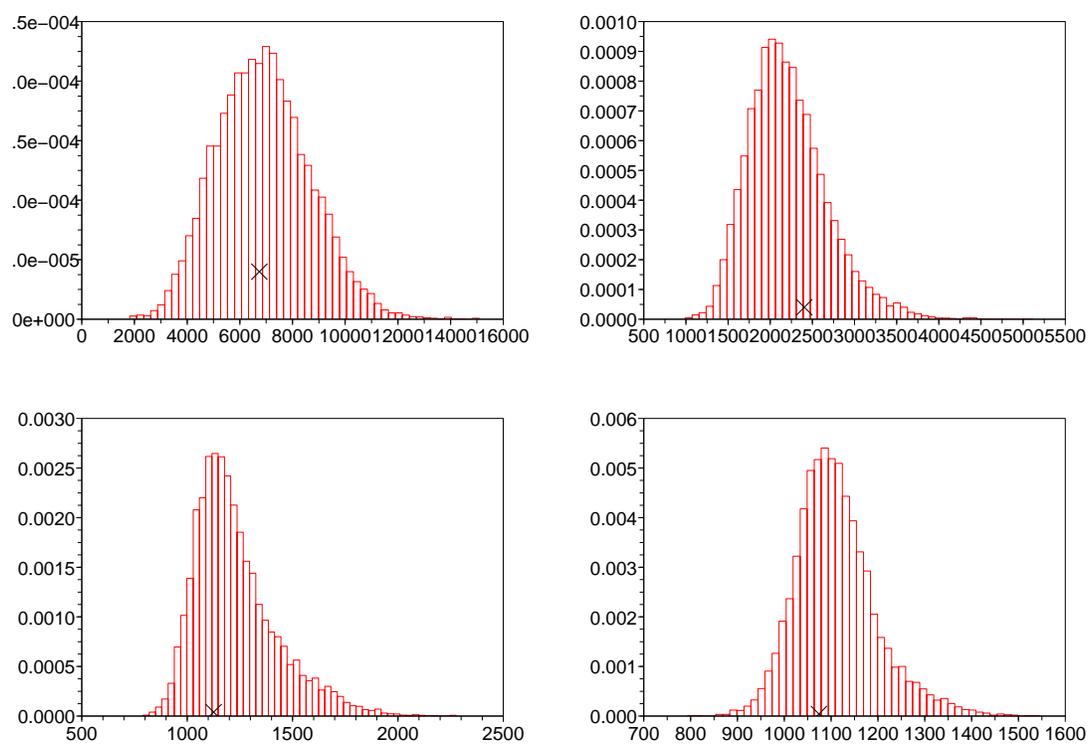


FIG. 4.12 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 1997$ (modèle densité-dépendant).

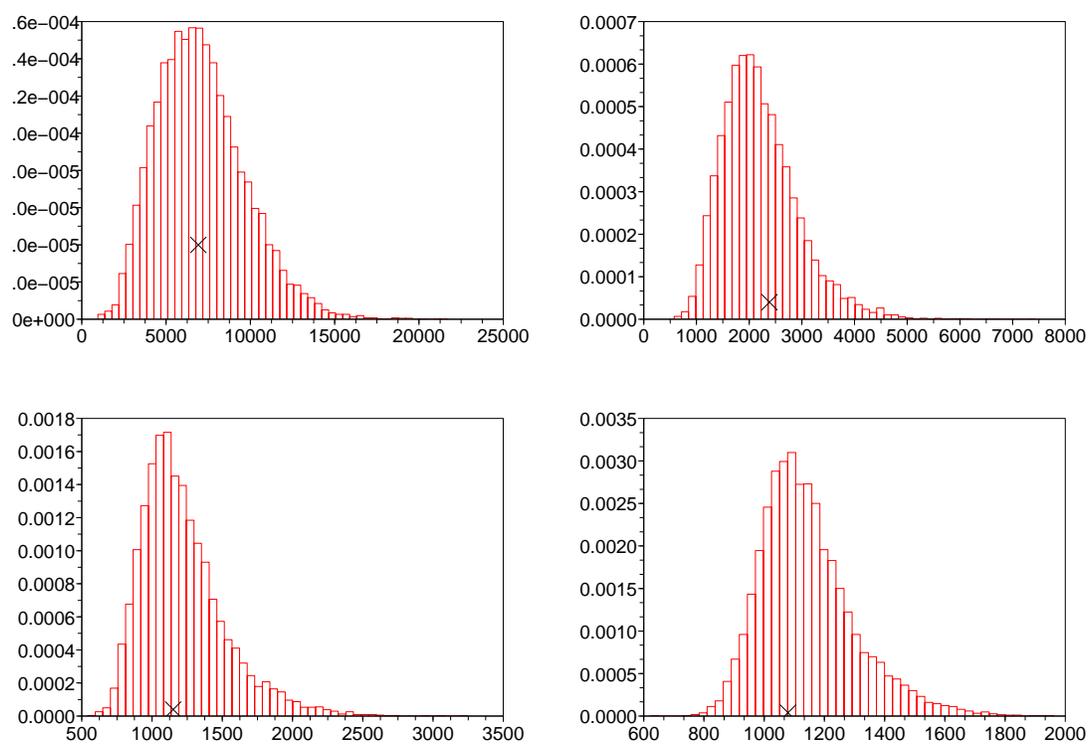


FIG. 4.13 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 1999$ (modèle densité-dépendant).

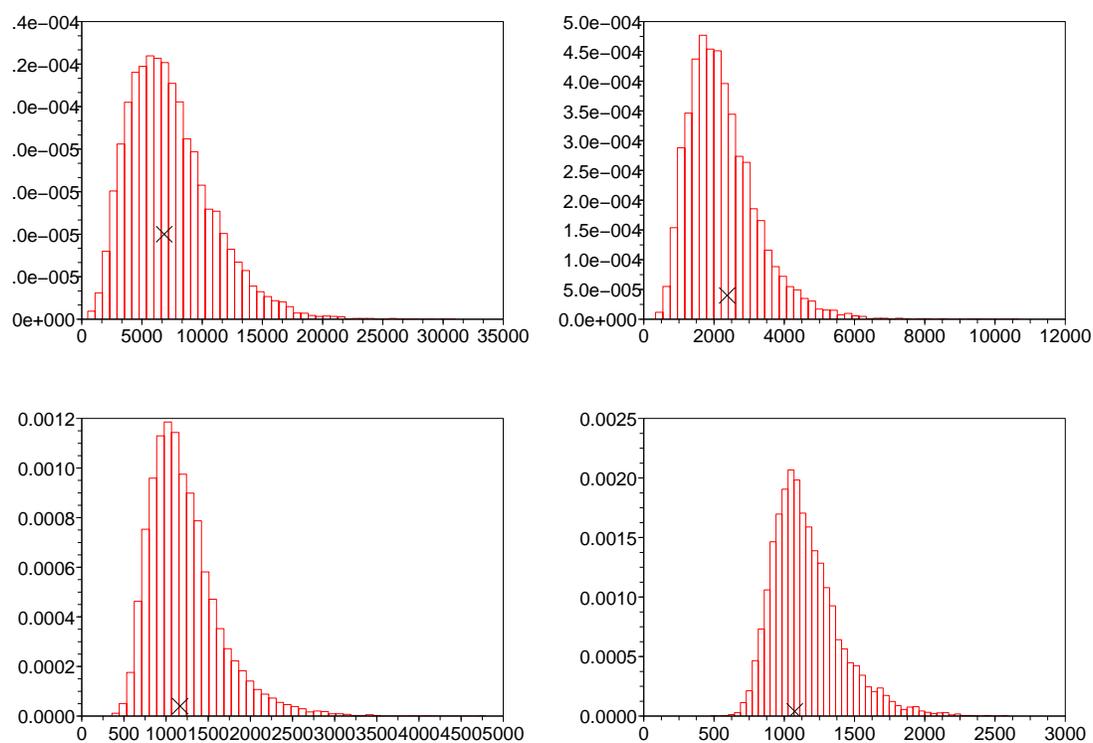


FIG. 4.14 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 2001$ (modèle densité-dépendant).

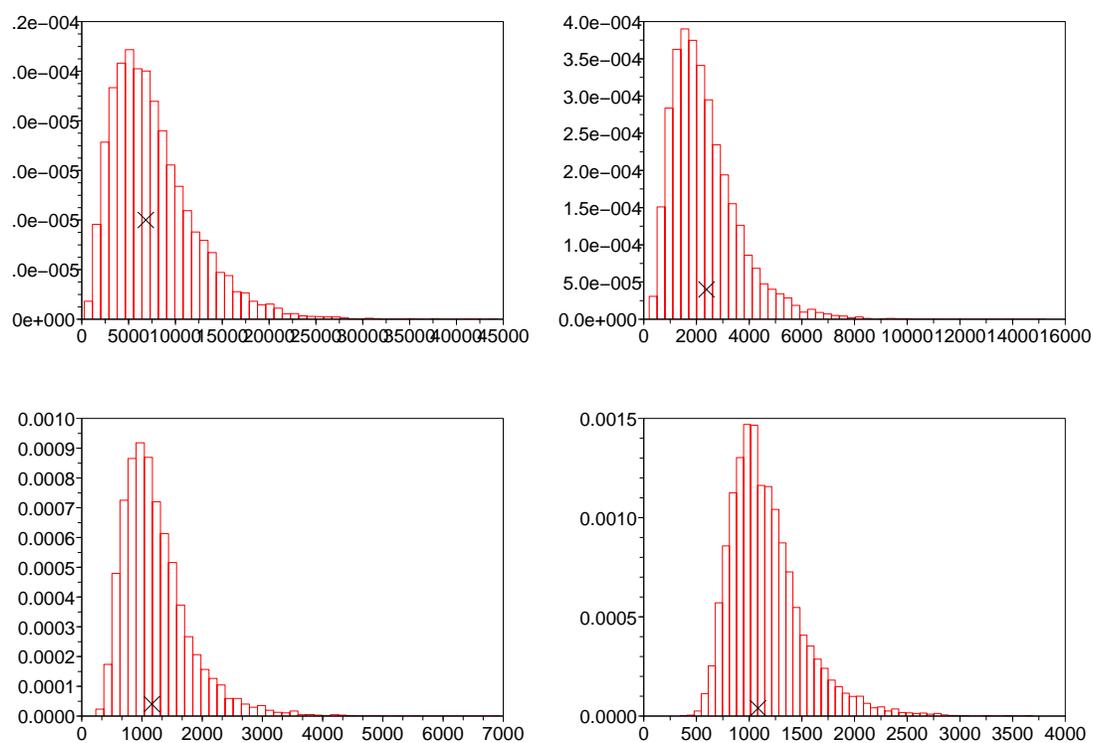


FIG. 4.15 – Loi a posteriori des effectifs N_t^i ($i = 1$ à 4) pour l'année $t = 2003$ (modèle densité-dépendant).

Conclusion

Notre travail a permis de proposer une adaptation des méthodes particulières à des modèles d'évolution de ressources renouvelables. Ces méthodes principalement développées dans la communauté du pistage peuvent donc se mettre en oeuvre dans le domaine de l'environnement. Elles mettent ainsi à disposition des écologues une approche alternative aux méthodes MCMC.

Bien que ces méthodes soient robustes et assez rapides, nous avons pu constater qu'il était possible de sensiblement les améliorer, notamment en améliorant la partie propagation des particules, soit au travers d'une fonction de proposition prenant en compte la nouvelle observation (ASIR), soit par l'intermédiaire d'une méthode MCMC (Resample-move). Ces algorithmes n'ont pas été utilisés en pratique dans le contexte habituel du filtrage particulaire car ils sont trop lents pour les applications classiques de poursuite. Dans le domaine de l'environnement le contexte est différent : nous disposons de suffisamment de temps entre deux observations consécutives. Notons que le resample-move est très prometteur, il combine en effet les deux approches (MCMC et filtrage particulaire).

La première application que nous avons traitée sur un modèle d'évolution de biomasse a permis de mettre en évidence la similarité des résultats entre les deux méthodes. L'obtention par deux méthodes différentes de résultats similaires conforte donc leurs validités.

L'approche que nous avons mise en place offre ainsi une alternative aux méthodes MCMC pour l'inférence sur les modèles environnementaux. De plus la mise en oeuvre est aussi simple que les méthodes MCMC mais le temps de calcul est beaucoup plus faible.

La deuxième application réalisée dans le domaine de la foresterie a permis de montrer que les méthodes particulières fonctionnent pour ce type de problème.

Le futur de cette première étude est multiple. Il est possible de pousser

le développement de logiciels en les mettant à la disposition du plus grand nombre au travers de plateforme logiciels comme R, Matlab ou Scilab. Il est aussi possible d'aborder un des problèmes les plus ardues du domaine, à savoir le contrôle du nombre de particules pour l'approximation particulaire. L'aspect MCMC se développe aussi considérablement au travers d'approche de chaînes en parallèles. Enfin, les liens entre filtrage particulaire et MCMC, toutes deux vues comme méthodes d'inférence bayésienne, sont encore à étudier.

Annexe A

Algorithmes

Voici les différents algorithmes utilisés ou mentionnés dans notre exposé.

A.1 Ellipse de confiance

La fonction qui suit va nous permettre de calculer et de dessiner une ellipse de confiance à 90% pour une loi gaussienne de dimension 2, cet algorithme est notamment utilisé dans les exemples 2.2.4 et 2.2.6.

```
function [ellipse]=f_plot_ellipse(fmean, fcov, npoint, R, temps)
    [vec,val]=spec(fcov);
    t=linspace(0, 2 * (%pi),npoint);
    ellipse=vec*sqrt(val*R)*[cos(t); sin(t)];
    ellipse=ellipse+ mtlb_repmat([fmean(1); fmean(2)],1,npoint);
    xset('color', 1);
    plot2d(ellipse(1, :), ellipse(2, :), 1);
endfunction
```

A.2 Redistribution des résidus

Cet algorithme présente l'autre technique de redistribution des particules, elle est mentionnée dans le paragraphe 3.2.

```
function ind=f_resample_residual(weights)
    np=length(weights);
    off=floor(np*weights);
```

```
weights=weights-(1/np)*off;
nnp=np-sum(off);
weights=weights/sum(weights);
weights=cumsum(weights);
weights=[0,weights];
for k=1 :nnp
    a=rand();
    i=sum(a>weights);
    off(i)=off(i)+1;
end
ind=f_resample_indice(off);
endfunction
```

Le tableau off de la fonction précédente nous renvoie dans $off(i)$ le nombre de descendants pour chacune des particules i , il est donc nécessaire de retourner les indices des particules sélectionnées, c'est ce qui est fait par la fonction $f_resample_indice$ qui calcule le tableau $ind(1 : N)$ des indices des particules sélectionnées

```
function ind=f_resample_indice(ftable)
    fftable=cumsum(ftable+1);
    a=zeros(1, fftable(length(fftable)));
    a(fftable)=1;
    aa=cumsum(a);
    ind=aa(ã)+1;
endfunction
```

Annexe B

Données de pêche

Voici les indices d'abondance et les captures récoltés par la Délégation à la Surveillance Pêche et au Contrôle en Mer (DSPCM) entre 1971 et 2004. Ce sont ces données, transmises par Étienne Rivot, que nous avons utilisées dans notre application.

	IA new	Captures
1971	3,40	38200
1972	3,44	36600
1973	3,13	29200
1974	2,46	36600
1975	1,99	39900
1976	2,11	52900
1977	3,14	36600
1978	3,04	23300
1979	1,84	13400
1980	2,62	12500
1981	1,24	32200
1982	2,10	39100
1983	1,67	29000
1984	1,79	29500
1985	1,41	38000
1986	0,90	45600
1987	1,37	45700
1988	1,65	36700
1989	1,15	33500
1990	1,10	22840
1991	1,24	28237
1992	1,66	42208
1993	1,22	42182
1994	0,98	33883
1995	0,75	29992
1996	0,53	31631
1997	0,53	19448
1998	0,61	15456
1999	0,79	22453
2000	0,87	31389
2001	0,52	26531
2002	0,43	23556
2003	0,52	22700
2004	0,66	21181

Bibliographie

- [1] S. Gourellet-fleuri et N. Picard A. Franc. *Forêts hétérogènes*. Oxford University Press, Englewood Cliffs, NJ, 1999.
- [2] M. Sanjeev Arulampalam, Simon Maksell, Neil J. Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, SP-50(2 (Special issue on Monte Carlo Methods for Statistical Signal Processing)) :174–188, February 2002.
- [3] Fabien Campillo and Vivien Rossi. Convolution particle filters for parameter estimation in general state-space models. Research Report RR-5939, INRIA, 06 2006. <http://hal.inria.fr/inria-00081956>.
- [4] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, 2006.
- [5] Arnaud Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR310, Department of Engineering, University of Cambridge, February 1998. <http://www-sigproc.eng.cam.ac.uk/~ad2/tr310.ps>.
- [6] Arnaud Doucet, Nando de Freitas, and Neil J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
- [7] Arthur Gelb, editor. *Applied Optimal Estimation*. MIT Press, 1974.
- [8] Walter R. Gilks and Carlo Berzuini. Following a moving target — Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B*, 63(1) :127–146, 2001.
- [9] Simon J. Godsill, Olivier Cappé, and Eric Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. In *To appear in IEEE Proceedings*, 2007.

-
- [10] Neil J. Gordon, David J. Salmond, and Adrian F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings, Part F*, 140(2) :107–113, April 1993.
 - [11] Peter S. Maybeck. *Stochastic Models, Estimation and Control*. Academic Press, 1979. 3 volumes (from 1979 to 1982).
 - [12] Michael K. Pitt and Neil Shephard. Auxiliary variable based particle filters. In Arnaud Doucet, Nando de Freitas, and Neil J. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, chapter 13, pages 273–293. Springer-Verlag, New York, 2001.
 - [13] Terrance J. Quinn and Richard B. Deriso. *Quantitative Fishery Dynamics*. Oxford, 1999.
 - [14] A. W. R.Kass and J. A. Wellner. MCMC for filtering via laplace approximation. 2005.
 - [15] Mélanie Zetlaoui. *Aspects statistiques de la stabilité en dynamique des populations : application au modèle de Usher en foresterie*. Thèse de Doctorat, Université de Paris 11, Paris, December 2006.