



Université Grenoble Alpes

Master 2 Mathématiques et Applications, parcours Statistique et Science des
Données (SSD)

RAPPORT DE PROJET TUTORÉ

Qui a une dent contre les narvals ?

Yanis BEN BELGACEM, Vadim BERTRAND, Angélique SAILLET

Encadrés par
Adeline LECLERCQ SAMSON
Frédéric AUDRA

Sommaire

1 Contexte	3
2 Effet des perturbations humaines sur l'alimentation des narvals	4
2.1 Modélisation de l'effet de l'exposition sur le taux d'émission	4
2.1.1 Processus de Poisson	4
2.1.2 Modèle linéaire généralisé	5
2.1.2.1 Effet de la profondeur	5
2.1.2.2 Effet de l'exposition	6
2.1.3 Non-indépendance des observations	6
2.1.3.1 Utilisation de modèles mixtes	6
2.1.3.2 Caractère autorégressif du processus	6
2.1.4 Effet de médiation de la profondeur	7
2.1.5 Intervalles de confiance	8
2.1.5.1 Construction par approche Monte-Carlo	8
2.1.5.2 Estimation de bandes de prédiction via la méthode Delta	10
2.2 Résultats	10
2.2.1 Note sur le temps d'ajustement des modèles	10
2.2.2 Recherche de la mémoire optimale	11
2.2.3 Régression double bi-exponentielle sur les coefficients autorégressifs	12
2.2.4 Impact de l'exposition aux perturbations sur le taux d'émission de buzz	12
2.2.5 Intervalles de confiance	14
2.2.5.1 Coefficients d'exposition	14
2.2.5.2 Pourcentage du taux normal d'émission de buzz	24
2.3 Conclusion	25

3 Estimation de la durée de vie des narvals grâce aux sillons de leur dent	26
3.1 Modèle sinusoïdal	27
3.1.1 Identifiabilité	28
3.2 Estimation des paramètres à partir d'un algorithme SAEM	30
3.2.1 Algorithme EM	31
3.2.2 Simulation de ξ_x	32
3.2.2.1 Algorithme MCMC	32
3.2.2.2 Algorithme SMC	33
3.2.3 Algorithme SAEM	34
3.3 Simulation & résultats	37
3.3.1 Estimation de ξ_x par MCMC	37
3.3.2 Estimation de ξ_x par SMC	39
3.3.3 Estimation de θ par SAEM	40
3.3.3.1 Avec une étape MCMC	40
3.3.3.2 Avec une étape SMC	43
3.3.3.3 Plan d'expérience	46
3.4 Conclusion	48
Références	50

Contexte

Le narval est une espèce de cétacés vivant dans l'océan Arctique, autour du Groenland. Le narval peut atteindre 5 mètres de long et un poids de 1600 kilogrammes et possède une longue durée de vie de 50 ans en moyenne. Il est surnommé licorne des mers en raison de sa dent ressemblant à une corne pouvant mesurer jusqu'à 3 mètres de long.

Dans le cadre du projet tutoré de notre 2^{ème} de Master Statistique et Science des Données nous nous sommes intéressés à deux questions concernant cet animal :

- l'impact des perturbations humaines sur le comportement des narvals,
- l'estimation de la durée de vie des narvals à partir des sillons de leur dent.

Ce travail s'inscrit dans une collaboration du Laboratoire Jean Kuntzmann (LJK) de Grenoble avec l'Université de Copenhague et l'Institut groenlandais des ressources naturelles. Nous tenons à fortement remercier notre tutrice de projet, Madame Adeline Leclercq Samson, chercheuse au LJK et grandement impliquée dans cette collaboration, de nous avoir accompagnés et guidés tout au long du projet, mais aussi d'avoir fait en sorte qu'il nous soit le plus intéressant possible en nous impliquant sur le choix de certains axes à investiguer.

Effet des perturbations humaines sur l'alimentation des narvals

Les narvals sont des baleines vivant toute l'année au Groenland. Le réchauffement climatique favorise le recul des glaces sur le territoire groenlandais et ses côtes. Cela ouvre la porte au développement d'activités humaines au Groenland, et notamment des activités minières. Les biologistes de l'Institut groenlandais des ressources naturelles se questionnent sur les effets potentiels engendrés par la présence humaine sur les comportements des narvals.

Afin d'anticiper ces possibles modifications de comportements une étude a été conduite pendant plusieurs mois en 2018 [1]. Dans ce cadre, 6 narvals ont été équipées de capteurs permettant d'enregistrer leur profondeur de plongée, leur localisation et les sons qu'elles émettent. Les baleines ont été laissées libres de toutes perturbations pendant plusieurs jours avant d'y être exposées. Les perturbations ont pris la forme de coups de fusil tirés dans l'eau depuis un bateau afin d'imiter les ondes émises par des activités minières.

Lorsqu'elles se nourrissent, les narvals émettent des sons spécifiques appelés “buzz”. À partir des sons collectés il est donc possible de déterminer quand ces baleines sont en train de manger. La distance séparant les baleines du bateau émettant une perturbation peut être calculée grâce aux puces GPS placées sur les narvals. Ainsi, nous pouvons modéliser l'effet de l'exposition des perturbations humaines sur l'émission de buzz des narvals et donc indirectement sur leur alimentation.

2.1 Modélisation de l'effet de l'exposition sur le taux d'émission

2.1.1 Processus de Poisson

Les données sont collectées toutes les secondes, ainsi nous disposons des temps de collecte T_j et $T_{j+1} = T_j + \Delta$ avec $T_0 = 0$ et $\Delta = 1$ seconde. On peut considérer $N(t)$, le nombre de buzz émis entre le début de la collecte et un instant t , comme un processus stochastique pour lequel $N(0) = 0$ et donnant $N(b) - N(a)$ le nombre de buzz émis dans l'intervalle $[a, b]$.

Les processus de comptage sont classiquement représentés par un processus de Poisson dont les accroissements sont indépendants et suivent une loi de Poisson : $N(t + \Delta) - N(t) \sim Pois$. Dans le

cas où l'intensité λ du processus dépend de t , on parle de processus non-homogène et :

$$N(t + \Delta) - N(t) \sim Pois\left(\int_t^{t+\Delta} \lambda(u)du\right)$$

On a donc que :

$$\mathbb{P}(N(t + \Delta) - N(t) = k) = e^{-\int_t^{t+\Delta} \lambda(u)du} \frac{\left(\int_t^{t+\Delta} \lambda(u)du\right)^k}{k!}$$

En utilisant le développement limité de l'exponentielle et en considérant $\lambda(t)$ constante sur $]t, t + \Delta]$ on obtient :

$$\begin{aligned}\mathbb{P}(N(t + \Delta) - N(t) = 0) &= 1 - \lambda(t)\Delta + o(\Delta) \\ \mathbb{P}(N(t + \Delta) - N(t) = 1) &= \lambda(t)\Delta + o(\Delta) \\ \mathbb{P}(N(t + \Delta) - N(t) \geq 2) &= o(\Delta)\end{aligned}$$

Ainsi, si Δ est suffisamment petit (de l'ordre de la seconde par exemple), $o(\Delta)$ est négligeable et le processus $Y(t) := N(t + \Delta) - N(t)$ prend uniquement comme valeurs 0 ou 1. On peut alors ramener notre processus de comptage à un processus de Bernoulli : $Y(t) \sim \mathcal{B}(\lambda(t)\Delta)$, ce qui implique :

$$\mathbb{E}(Y(t)) = \mathbb{P}(Y(t) = 1) = \lambda(t)\Delta \underset{\Delta=1}{=} \lambda(t)$$

2.1.2 Modèle linéaire généralisé

Nous cherchons à estimer le lien entre l'intensité d'émission de buzz et plusieurs covariables, dont le niveau d'exposition aux perturbations. Nous avons vu dans la section précédente que les variables aléatoires $Y(t)$ de notre processus de comptage sont à valeurs dans 0, 1 et que $\mathbb{P}(Y(t) = 1) = \lambda(t)$. En se plaçant dans le cadre des modèles linéaires généralisés (GLM) il est donc possible d'exprimer la probabilité $\lambda(t)$ que $Y(t) = 1$ selon un vecteur de covariables $Z(t)$ (détaillées dans la suite de cette section) en utilisant le logarithme comme fonction de lien :

$$\log(\lambda(t)) = \beta_0 + Z(t)^T \beta_Z$$

avec β_Z le vecteur de paramètres associés au vecteur $Z(t)$.

2.1.2.1 Effet de la profondeur

Pour se nourrir, les narvals doivent plonger profondément (plusieurs centaines de mètres), alors que le reste du temps elles restent "proches" (quelques dizaines de mètres) de la surface. Il faut donc inclure au modèle la covariable de profondeur à laquelle se trouvent les baleines quand elles émettent ou non des buzz. La relation entre l'émission de buzz et la profondeur n'étant pas linéaire, la profondeur a été remplacée par une spline cubique naturelle ayant pour noeuds les quantiles 1/3 et 2/3.

2.1.2.2 Effet de l'exposition

Le niveau d'exposition aux perturbations est représenté par l'inverse de la distance séparant la baleine du bateau quand un coup de feu est tiré. De même que pour la profondeur, la non-linéarité de la relation entre le niveau d'exposition et le taux d'émission de buzz est représentée par l'utilisation d'une spline cubique naturelle dont les noeuds sont les quantiles 1/3 et 2/3 des niveaux d'exposition.

2.1.3 Non-indépendance des observations

L'utilisation d'un processus de Poisson pour modéliser nos données de comptage implique l'indépendance des observations $Y(t)$. En pratique ce n'est pas le cas, aussi il faut modifier le modèle défini précédemment pour tenter de compenser cette absence d'indépendance.

2.1.3.1 Utilisation de modèles mixtes

Les données que nous utilisons correspondent à plusieurs individus, nous avons donc plusieurs observations par individu et celles-ci ne sont pas indépendantes. Pour palier ce défaut de modélisation et tenir compte de la spécificité des individus, nous utilisons des modèles mixtes en ajoutant un effet aléatoire b_i sur l'ordonnée à l'origine :

$$\log(\lambda_i(t)) = \beta_0 + \textcolor{blue}{b_i} + \text{spline}(D_i(t))\beta_{D_{1:3}} + \text{spline}(E_i(t))\beta_{E_{1:3}}$$

où i dénote l'individu i et $b_i \sim \mathcal{N}(0, \sigma^2)$ l'effet aléatoire sur cet individu, σ^2 étant la variance inter-individuelle.

2.1.3.2 Caractère autorégressif du processus

De plus, l'émission d'un buzz à un instant t est corrélé à l'émission ou non de buzz aux instants précédents ; cet effet mémoire doit donc être intégré au modèle pour tenir compte de la dépendance des $Y_i(t)$. Pour cela nous introduisons K variables binaires d'autorégression codant l'émission d'un buzz aux instants $t - k$, $k \in \{1, \dots, K\}$. Le modèle résultant s'écrit alors :

$$\log(\lambda_i(t)) = \beta_0 + b_i + \text{spline}(D_i(t))\beta_{D_{1:3}} + \sum_{k=1}^K \alpha_k Y_i(t-k) + \text{spline}(E_i(t))\beta_{E_{1:3}}$$

Cette approche demande de fixer une mémoire maximale, et ainsi la valeur de K . Pour choisir la mémoire maximale optimale, nous avons fait varier K et utilisé le BIC comme mesure de la qualité des différents modèles correspondants sans inclure l'effet de l'exposition :

$$\log(\lambda_i(t)) = \beta_{D_0} + b_i + \text{spline}(D_i(t))\beta_{D_{1:3}} + \sum_{k=1}^K \alpha_k Y_i(t-k) \quad (2.1)$$

Nous choisissons la mémoire maximale K_{opt} du modèle minimisant ce critère. Pour éviter de parcourir tout l'ensemble $\{K_{min}, \dots, K_{max}\}$, nous avons utilisé la démarche proposée par l'Algorithme 1 permettant de restreindre l'ensemble de recherche au fur et à mesure que l'on s'approche de K_{opt} .

Algorithme 1 Réduction progressive de l'ensemble de recherche de K_{opt} .

```

 $from_k \leftarrow K_{min}; to_k \leftarrow K_{max}$                                 ▷ Bornes de l'intervalle de recherche
 $M \leftarrow 10$                                                                ▷ Nombre d'éléments évalués dans l'intervalle
while ( $to_k - from_k > 2$ ) do ▷ On s'arrête quand on évalué l'ensemble du voisinage du minimum
     $K_{1:M} \leftarrow NA$ 
     $BIC_{1:M} \leftarrow NA$ 
    for  $i \in \{1, \dots, M\}$  do
         $K_i \leftarrow from_k + (i - 1) * \lfloor \frac{to_k - from_k}{M-1} \rfloor$           ▷ Découpage en  $M$  éléments équidistants
         $\mathcal{M} \leftarrow$  ajustement du modèle de l'Equation (2.1) avec  $K_i$  éléments mémoire
         $BIC_i \leftarrow$  calcul du BIC de  $\mathcal{M}$ 
    end for
     $i_{opt} \leftarrow argmin BIC$ 
     $K_{opt} \leftarrow K_{i_{opt}}$ 
     $from_k \leftarrow K_{i_{opt}-1}; to_k \leftarrow K_{i_{opt}+1}$           ▷ Mise à jour des bornes en encadrant le minimum
end while

```

Le nombre de composants autorégressifs pouvant être grand, il faut les lier aux α_k avec un modèle de régression. Pour cela, nous avons utilisé une régression double bi-exponentielle :

$$BiExp(lag) = A_1 e^{-e^{lrc_1} lag} + A_2 e^{-e^{lrc_2} lag}$$

Cela permet de réduire le nombre de coefficients de K_{opt} à 4, ce qui est doublement bénéfique : le temps d'ajustement des modèles est grandement réduit et lors de la construction des intervalles de confiance de nos coefficients, l'accumulation des variances est limitée.

Le modèle complet s'exprime donc ainsi :

$$\log(\lambda_i(t)) = \beta_0 + b_i + spline(D_i(t))\beta_{D_{1:3}} + \sum_{k=1}^{K_{opt}} (A_1 e^{-e^{lrc_1} k} + A_2 e^{-e^{lrc_2} k}) Y_i(t-k) + spline(E_i(t))\beta_{E_{1:3}}$$

2.1.4 Effet de médiation de la profondeur

Il est possible que l'exposition à des perturbations conduisent les narvals à :

1. émettre moins de buzz,
2. moins plonger ou plonger moins profondément.

Mais il existe également un lien entre la profondeur d'immersion des baleines et leur production de buzz. Aussi, il se peut que l'exposition ait un lien direct sur l'émission de buzz et un lien indirect via son effet sur la profondeur.

Afin de représenter uniquement le lien direct, les coefficients des covariables autres que l'exposition sont estimés sans inclure celle-ci à partir des observations effectuées sans soumettre les animaux à des perturbations :

$$\log(\lambda_i(t)) = \beta_{D_0} + b_i + spline(D_i(t))\beta_{D_{1:3}} + \sum_{k=1}^{K_{opt}} A_1 e^{-e^{lrc_1} k} + A_2 e^{-e^{lrc_2} k} Y_i(t-k) \quad (2.2)$$

Nous obtenons donc les estimations $\widehat{\beta_D}$, $\widehat{A_{1:2}}$, $\widehat{lrc_{1:2}}$ qui sont ensuite injectées dans le modèle complet au moyen d'un terme d'"offset" (non réestimé) :

$$offset_i(t) = \widehat{\beta_{D_0}} + spline(D_i(t))\widehat{\beta_{D_{1:3}}} + \sum_{k=1}^{K_{opt}} (\widehat{A_1}e^{-e^{\widehat{lrc_1}}k} + \widehat{A_2}e^{-e^{\widehat{lrc_2}}k})Y_i(t-k) \quad (2.3)$$

Le modèle incluant l'exposition est donc reformulé ainsi :

$$\log(\lambda_i(t)) = \beta_{E_0} + b_i + offset_i(t) + spline(E_i(t))\beta_{E_{1:3}} \quad (2.4)$$

Et les paramètres estimés $\widehat{\beta_E}$ permettront d'évaluer l'effet de l'exposition par rapport à des conditions "normales".

2.1.5 Intervalles de confiance

2.1.5.1 Construction par approche Monte-Carlo

Après avoir estimé les coefficients associés à l'exposition aux perturbations, nous souhaitons construire les intervalles de confiance de ces estimations. L'approche classique de calcul des intervalles de confiance se basant sur la seule variance estimée des coefficients d'exposition donnerait ici des résultats incorrects, la variance des coefficients de profondeur et d'autorégression ne serait alors pas prise en compte car tuée par l'utilisation de l'offset.

2.1.5.1.1 Utilisation de lois normales univariées Dans un premier temps, nous avons donc ajusté le modèle sans exposition décrit par l'Equation (2.2) afin d'obtenir la moyenne et la variance empiriques de ses coefficients. Nous avons ensuite répété le tirage des coefficients sans exposition selon 8 lois normales univariées paramétrées par leurs statistiques empiriques ; calculé le terme d'offset selon l'Equation (2.3) à partir des réalisations obtenues ; et estimé les coefficients d'exposition $\widehat{\beta_E^k}$ du modèle correspondant à l'Equation (2.4). Les coefficients $\widehat{\beta_E^k}$ estimés à chaque itération constituent ainsi un échantillon dont nous utilisons les quantiles empiriques $\alpha/2$ et $1-\alpha/2$ comme bornes de l'intervalle de confiance au niveau α de chacun des coefficients d'exposition.

2.1.5.1.2 Utilisation de lois normales multivariées Le tirage selon des lois normales univariées implique que nous ne considérons pas la covariance existante entre les estimations des coefficients $\widehat{\beta_D}$, $\widehat{A_{1:2}}$ et $\widehat{lrc_{1:2}}$. Afin d'y remédier nous avons employé un tirage suivant 2 lois normales multivariées de dimension 4 (une pour les coefficients $\widehat{\beta_D}$ et une pour $\widehat{A_{1:2}}$ et $\widehat{lrc_{1:2}}$, dont le vecteur de moyenne est toujours constitué des moyennes empiriques des coefficients sans exposition, mais dont les matrices de variance-covariance ne sont pas diagonales).

2.1.5.1.3 Variance-covariance des coefficients autorégressifs La matrice de variance-covariance permettant de tirer les coefficients $\widehat{A_{1:2}}$ et $\widehat{lrc_{1:2}}$ était construite à partir des estimations de la régression double bi-exponentielle, ce qui signifie que nous ne captions pas directement la variabilité du phénomène autorégressif mais plutôt celle de la régression double bi-exponentielle.

Pour palier cette approximation, nous avons recalculé la matrice de variance-covariance des coefficients $\widehat{A}_{1:2}$ et $\widehat{lrc}_{1:2}$ par une autre procédure Monte-Carlo en répétant l'ajustement de la régression double bi-exponentielle pour des coefficients mémoire tirés selon une loi normale multivariée. Cette approche est décrite par l'Algorithme 2.

Algorithme 2 Procédure Monte-Carlo d'obtention des intervalles de confiance.

```

 $\mathcal{M}_0 \leftarrow$  ajustement du modèle correspondant à l'Equation (2.1)
 $\mu_D, \Sigma_D, \mu_{AR}, \Sigma_{AR} \leftarrow$  estimation à partir de  $\mathcal{M}_0$ 
 $K \leftarrow 1000$  ▷ Nombre de répétitions des procédures
▷ 1ère procédure
 $\widehat{\beta}_{A_{1:2}}^{(1:K)} \leftarrow NA$ 
 $\widehat{\beta}_{lrc_{1:2}}^{(1:K)} \leftarrow NA$ 
for  $k \in \{1, \dots, K\}$  do
     $\hat{\alpha} \leftarrow \mathcal{N}(\mu_{AR}, \Sigma_{AR})$  ▷ Tirage des coefficients autorégressifs
     $\mathcal{M}_{exp} \leftarrow$  ajustement de la régression double bi-exponentielle utilisant  $\hat{\alpha}$ 
     $\widehat{\beta}_{A_{1:2}}^{(k)}, \widehat{\beta}_{lrc_{1:2}}^{(k)} \leftarrow$  coefficients de  $\mathcal{M}_{exp}$ 
end for
 $\mu_{exp}, \Sigma_{exp} \leftarrow$  calculés à partir de  $\widehat{\beta}_{A_{1:2}}, \widehat{\beta}_{lrc_{1:2}}$  ▷ 2ème procédure
 $\widehat{\beta}_E^{(1:K)} \leftarrow NA$ 
for  $k \in \{1, \dots, K\}$  do
     $\widehat{\beta}_D \leftarrow \mathcal{N}(\mu_D, \Sigma_D)$  ▷ Tirage des coefficients de profondeur
     $\widehat{A}_{1:2}, \widehat{lrc}_{1:2} \leftarrow \mathcal{N}(\mu_{exp}, \Sigma_{exp})$  ▷ Tirage des coefficients de la double bi-exponentielle
    calcul de l'offset selon l'Equation (2.3) avec  $\widehat{\beta}_D, \widehat{A}_{1:2}, \widehat{lrc}_{1:2}$ 
     $\mathcal{M} \leftarrow$  ajustement du modèle donné par l'Equation (2.4) en fixant l'offset
     $\widehat{\beta}_E^{(k)} \leftarrow$  coefficients de profondeur de  $\mathcal{M}$ 
end for
 $IC_\alpha \leftarrow [\widehat{q}_{\alpha/2}^{\widehat{\beta}_E}, \widehat{q}_{1-\alpha/2}^{\widehat{\beta}_E}]$  ▷ Estimation via les quantiles empiriques

```

2.1.5.1.4 Sans passer par la régression double bi-exponentielle

L'intérêt de l'utilisation de la régression double bi-exponentielle est de :

1. réduire le temps d'ajustement des modèles linéaires,
2. éviter d'accumuler les variances des 60 coefficients de mémoire.

En fixant les coefficients autorégressifs, avoir 4 ou 60 coefficients pour la mémoire n'importe plus ; et en utilisant une loi normale multivariée, nous devrions également ne plus accumuler directement les variances des coefficients. Nous pouvons donc envisager de nous passer de la régression double bi-exponentielle, et de tirer les 60 coefficients mémoire et les 4 coefficients de profondeur directement dans une seule loi normale multivariée dont les paramètres sont obtenus après ajustement du modèle sans exposition de l'Equation (2.1). Ainsi nous revenons à une seule procédure Monte-Carlo (la

deuxième dans l'Algorithme 2) et l'offset utilisé dans l'Equation (2.4) du modèle complet devient :

$$offset_i(t) = \widehat{\beta_{D_0}} + spline(D_i(t))\widehat{\beta_{D_{1:3}}} + \sum_{k=1}^{K_{opt}} \widehat{\alpha_k} Y_i(t-k)$$

2.1.5.2 Estimation de bandes de prédition via la méthode Delta

Une fois les coefficients d'exposition $\widehat{\beta_E}$ estimés nous pouvons prédire le taux d'émission de buzz sur l'ensemble d'un intervalle de niveau d'exposition aux perturbations. Afin d'associer une bande de confiance à la prédition moyenne nous avons appliqué la méthode Delta pour calculer la variance du taux d'émission.

Tout d'abord, d'après le Théorème Central Limite on a : $\sqrt{n}(\widehat{\beta} - \beta) \sim \mathcal{N}(0, \Sigma)$ avec $\widehat{\beta}$ les coefficients estimés du modèle et Σ leur variance.

Etant donné que nous utilisons un GLM avec un lien log, le taux d'émission est lié aux coefficients du modèle via la relation $\lambda(t) = \exp(X(t)\beta) := f(\beta)$. Le gradient de la fonction f vaut $\nabla f(\beta) = Xf(\beta)$.

Ainsi, quand on applique la méthode Delta on obtient :

$$\begin{aligned} \sqrt{n}(f(\widehat{\beta}) - f(\beta)) &\xrightarrow{\mathcal{L}} \mathcal{N}(0, \nabla f(\beta)^T \Sigma \nabla f(\beta)) \\ \Leftrightarrow \sqrt{n}(\widehat{\lambda(t)}) - \lambda(t) &\xrightarrow{\mathcal{L}} \mathcal{N}(0, \lambda(t)^2 X(t)^T \Sigma X(t)) \end{aligned}$$

donc

$$\frac{\sqrt{n}}{\widehat{\lambda(t)} \sqrt{X(t)^T \widehat{\Sigma} X(t)}} (\widehat{\lambda(t)}) - \lambda(t) \rightsquigarrow \mathcal{N}(0, 1)$$

On en déduit la bande de confiance au niveau α :

$$IC_\alpha(\lambda(t)) = [\widehat{\lambda(t)} \pm q_{1-\alpha/2}^{\mathcal{N}} \frac{\widehat{\lambda(t)} \sqrt{X(t)^T \widehat{\Sigma} X(t)}}{\sqrt{n}}]$$

où $q_{1-\alpha/2}^{\mathcal{N}}$ est le quantile $1 - \alpha/2$ de la loi normale centrée réduite.

2.2 Résultats

2.2.1 Note sur le temps d'ajustement des modèles

Le temps d'ajustement des modèles, et en particulier des modèles mixtes, augmente fortement lorsque que le nombre de paramètres à ajuster augmente.

Comme nous pouvons le voir sur la Figure 2.1 cette augmentation est linéaire pour les modèles classiques, alors que pour les modèles mixtes celle-ci est quadratique.

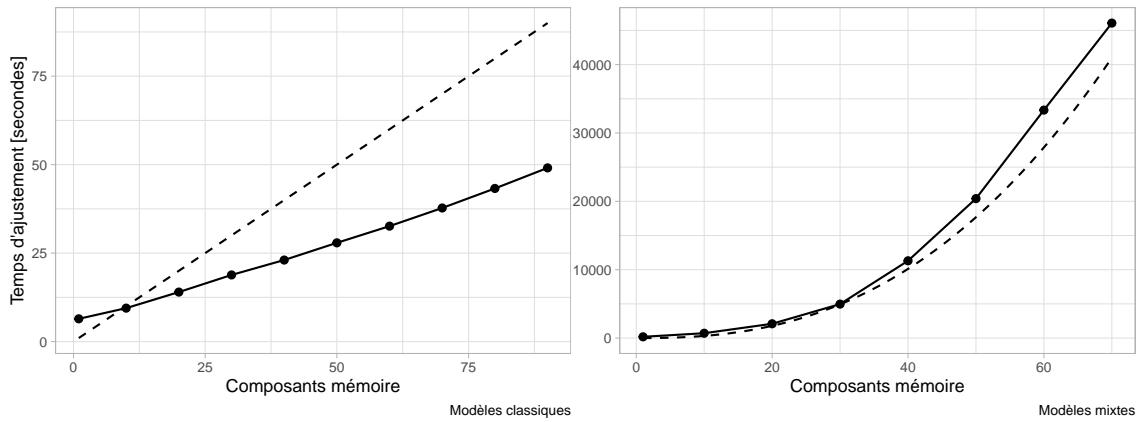


Figure 2.1: Courbes pleines : temps d'ajustement (en secondes) des modèles en fonction de la mémoire maximum ; courbes en pointillés : $f(x) = x$ à gauche et $f(x) = x^{2.5}$ à droite.

2.2.2 Recherche de la mémoire optimale

Le temps d'ajustement des modèles mixtes étant nettement plus important que ceux des modèles sans effets aléatoires, nous avons dans un premier temps exclu ces effets du modèle sans exposition pour estimer la mémoire optimale. Nous avons choisi pour la recherche $K_{min} = 1$, $K_{max} = 300$, $M = 10$. Nous obtenons une mémoire optimale de 60 secondes. Cela nous a permis de restreindre tout de suite l'ensemble de recherche initial à $K_{min} = 1$, $K_{max} = 300$ quand nous avons considéré le modèle incluant les effets aléatoires. De même que précédemment, la mémoire optimale est égale à 60 secondes. La Figure 2.2 permet de voir que dans les deux cas les optimums semblent bien correspondre à des minimums globaux.

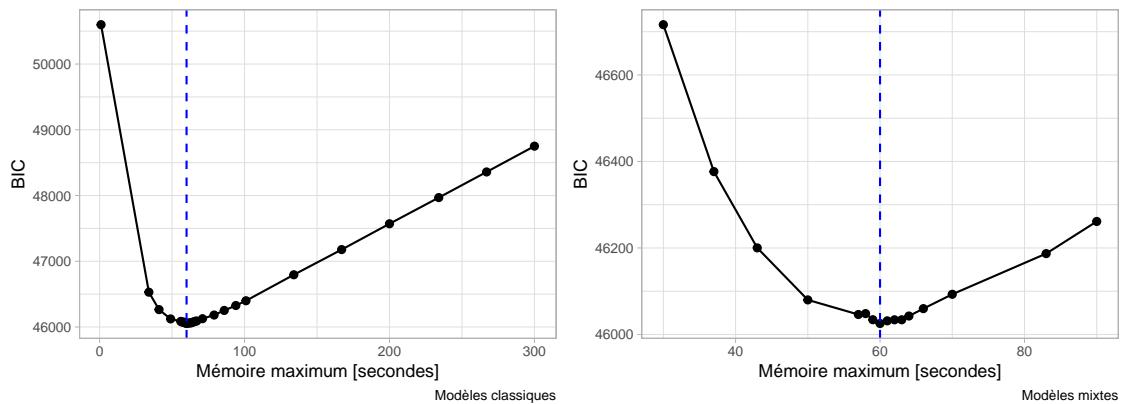


Figure 2.2: BIC en fonction de la mémoire maximum.

2.2.3 Régression double bi-exponentielle sur les coefficients autorégressifs

La Figure 2.3 permet de comparer la régression double bi-exponentielle et les composantes de la mémoire ajustées pour un décalage maximum de 60. Nous pouvons constater que l'ajustement par la double bi-exponentielle est très fidèle aux 60 coefficients initiaux.

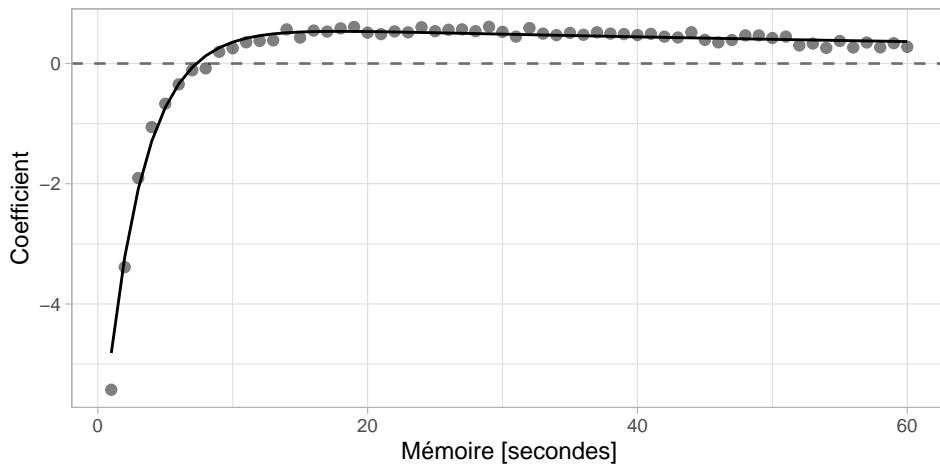


Figure 2.3: Régression double bi-exponentielle des coefficients autorégressifs.

Les 4 coefficients obtenus suite à la régression double bi-exponentielle sont présentés dans la Table 2.1. Nous pouvons noter que leurs erreurs standard sont faibles, ce qui est cohérent avec la validation visuelle de la régression.

Table 2.1: Coefficients autorégressifs obtenus par régression double bi-exponentielle

	moyenne	erreur standard
A_1	-7.741	0.329
lrc_1	-1.055	0.041
A_2	0.654	0.043
lrc_2	-4.633	0.188

2.2.4 Impact de l'exposition aux perturbations sur le taux d'émission de buzz

La Table 2.2 expose les coefficients associés à l'exposition du modèle mixte. Interpréter les valeurs des coefficients liés à des splines étant peu pertinent, nous préférons nous référer à la lecture de la Figure 2.4 sur laquelle nous représentons les estimations du modèle pour le taux d'émission

de buzz selon la distance avec le bateau. Il apparaît alors clairement que plus le bateau est loin, plus le taux d'émission de buzz est élevé.

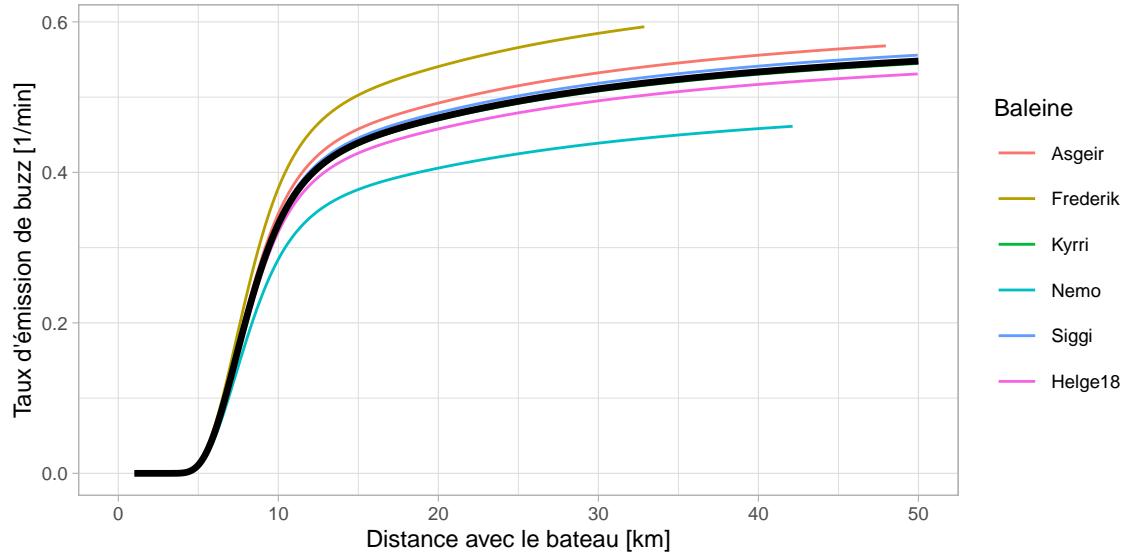


Figure 2.4: Evolution du taux d'émission de buzz selon la distance au bateau.

Table 2.2: Coefficients d'exposition.

	moyenne	erreur standard
β_{E_1}	-1.146	0.002
β_{E_2}	-58.872	0.054
β_{E_3}	-112.790	0.107

Nous voulons vérifier que notre processus de comptage est bien représenté par un processus de Poisson et que nous avons bien tenu compte de la corrélation entre ses accroissements. Pour cela, nous nous sommes intéressé aux résidus uniformes du modèle.

Pour vérifier l'absence de corrélation, nous avons représenté sur la Figure 2.5 l'autocorrélation des résidus en fonction du décalage temporel, ainsi que les résidus en fonction des résidus précédents. Nous remarquons qu'aucun motif de corrélation ne semble apparaître, ce qui vient valider notre approche pour inclure au modèle le lien naturel existant entre les émissions de buzz.

Le Q-Q plot de la Figure 2.5 confirme lui que le processus de Poisson est bien adapté aux données de comptage des buzz étant donné que les quantiles des résidus correspondent à ceux de la loi uniforme.

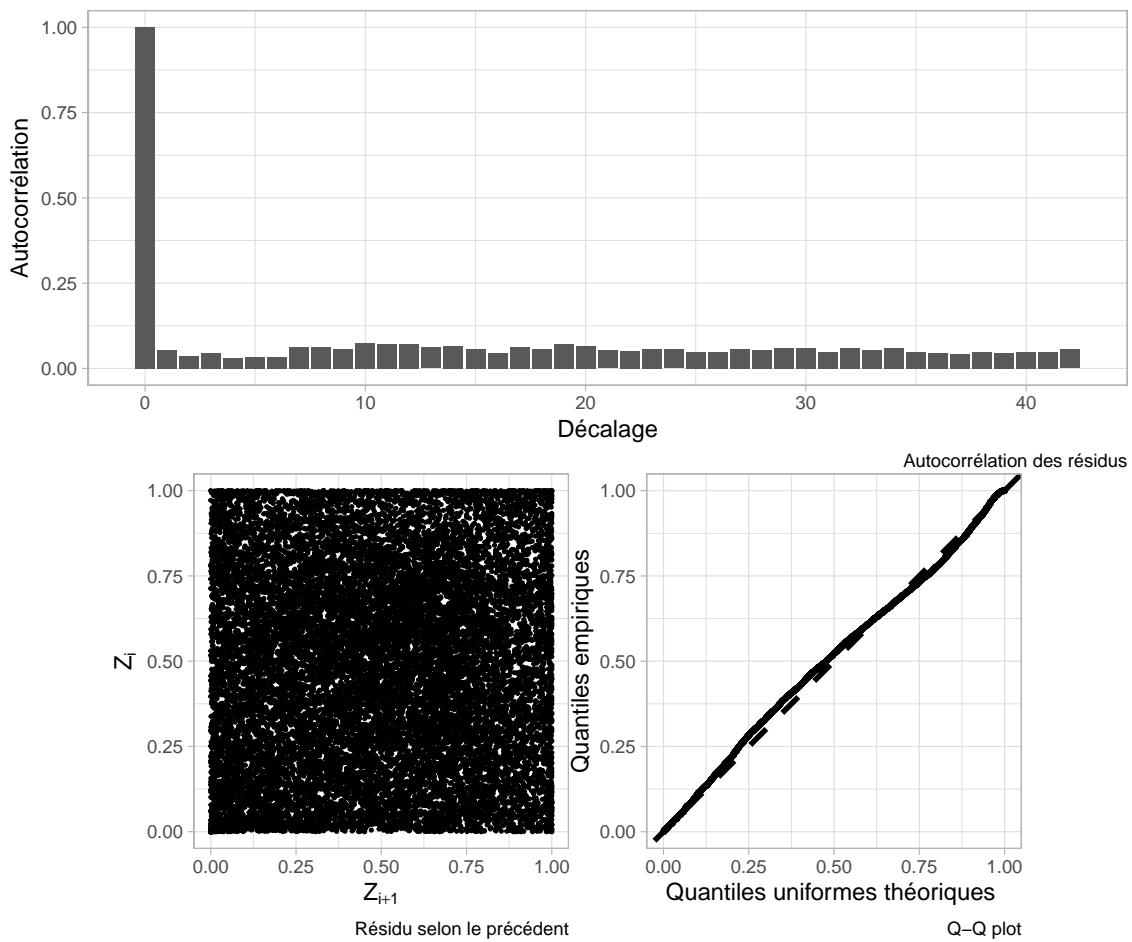


Figure 2.5: Validation graphique du modèle.

2.2.5 Intervalles de confiance

2.2.5.1 Coefficients d'exposition

2.2.5.1.1 Utilisation de lois normales univariées Comme détaillé dans la section 2.1.5.1.1, nous avons commencé par considérer que les coefficients autorégressifs et de profondeur suivaient chacun une loi normale centrée sur leur estimation moyenne et avec une variance égale au carré de leur erreur standard. Nous avons donc 4 lois normales univariées pour les coefficients de la régression double bi-exponentielle et 4 autres pour la spline sur la profondeur.

Sur la Figure 2.6 nous avons représenté les courbes des fonctions double bi-exponentielles ainsi générées. Nous pouvons voir que leurs allures semblent toujours suivre celle de la régression initiale.

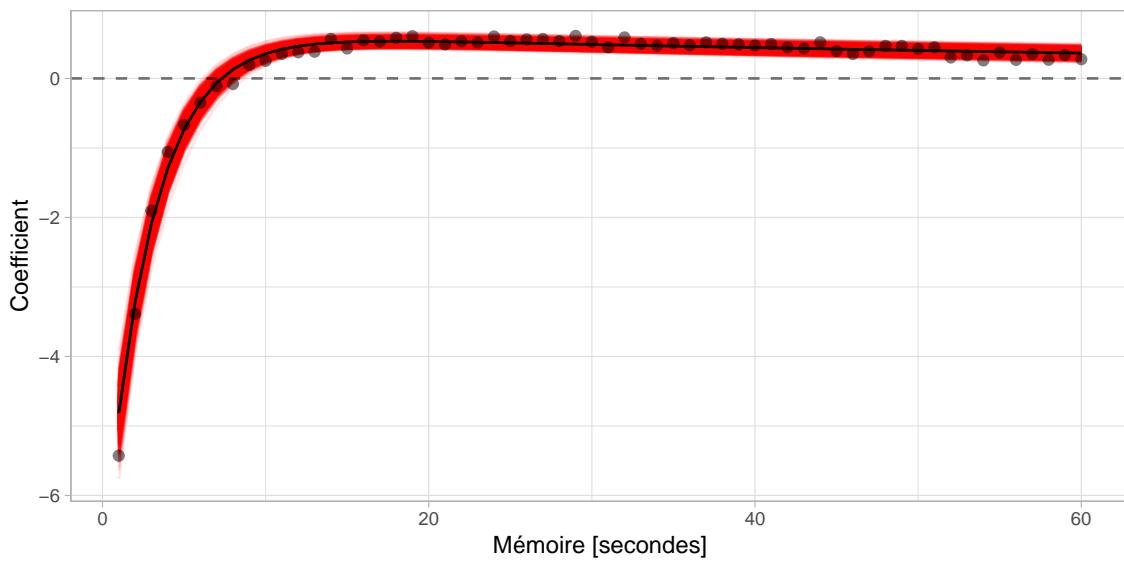


Figure 2.6: Variation des coefficients autorégressif selon des lois normales univariées.

Nous avons fait de même avec les coefficients de profondeur, et nous pouvons remarquer sur la Figure 2.7 que cette fois-ci certaines courbes s'éloignent sensiblement de l'interpolation moyenne.

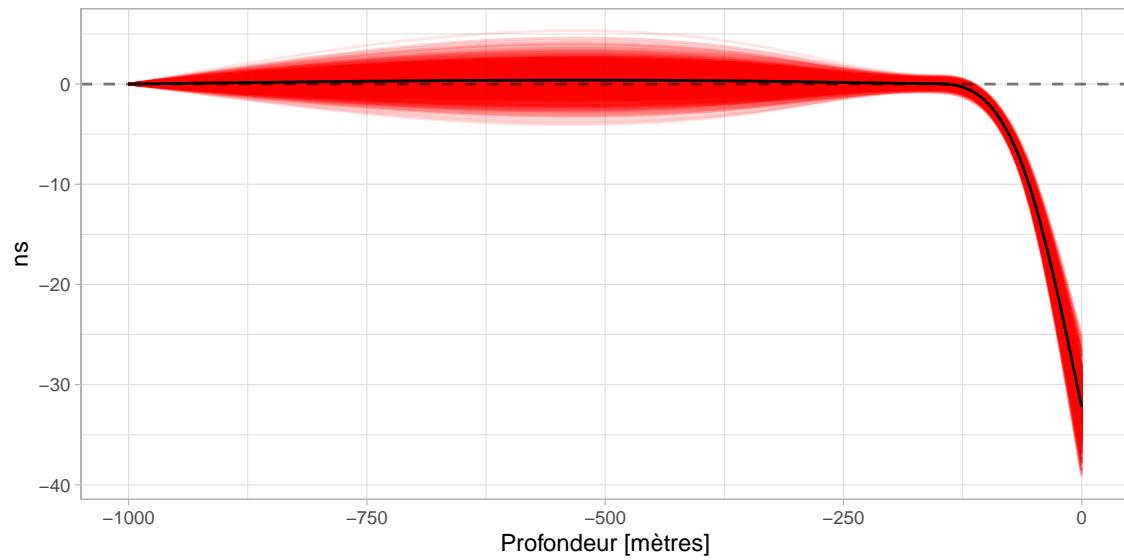


Figure 2.7: Variation des coefficients de profondeur selon des lois normales univariées.

Les intervalles de confiance calculés avec la procédure Monte-Carlo sont donnés sur la Table 2.3. Nous avons également affiché les valeurs médianes des intervalles de confiance calculés à chaque répétition sur la base de la variation des coefficients d'exposition uniquement. Il est flagrant que les intervalles Monte-Carlo sont bien plus larges et ne permettent en aucun cas de conclure sur un effet de l'exposition sur le taux d'émission de buzz.

Table 2.3: Intervalles de confiance dans le cas de normales univariées.

	Monte-Carlo		Erreur standard	
	inf	sup	inf	sup
β_0	-6.442	-2.968	-4.639	-4.494
β_{E_1}	-11.899	11.294	-1.430	-1.422
β_{E_2}	-152.959	1.048	-56.015	-55.803
β_{E_3}	-306.429	11.707	-106.847	-106.431

La Figure 2.8 donne une représentation graphique de ces intervalles (en rouge les “MC” et en bleu les “SE”), ainsi que des distributions des coefficients.

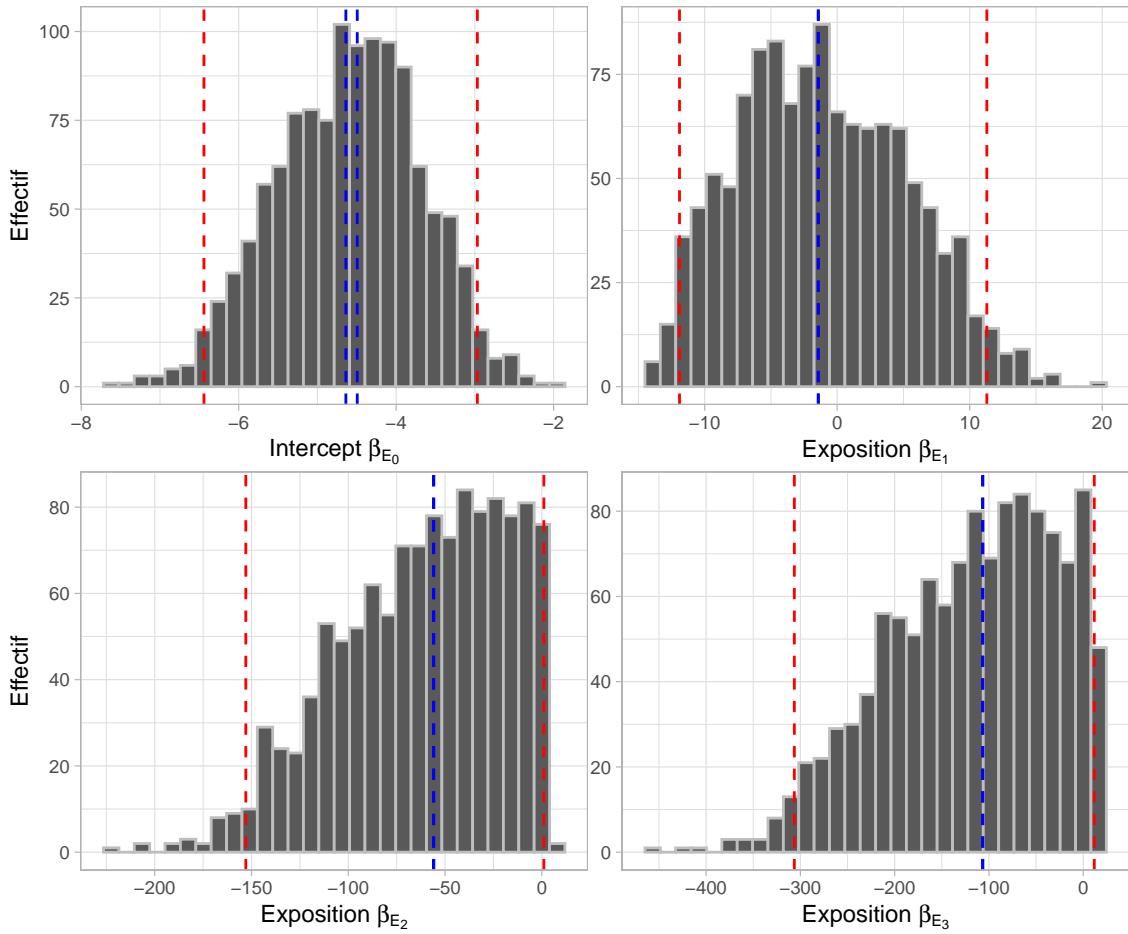


Figure 2.8: Intervalles de confiance dans le cas de normales univariées.

2.2.5.1.2 Utilisation de lois normales multivariées L'une des explications aux très larges intervalles de confiance observés dans la section précédente pourrait être que nous avons accumulé les variances des coefficients sans tenir compte des probables covariances existant entre les coefficients. Afin de corriger cela nous avons répété l'approche décrite précédemment, mais en tirant les coefficients dans 2 lois normales multivariées : une pour les coefficients de la régression double bi-exponentielle et une pour les coefficients de profondeur. Pour plus de détails sur l'obtention des paramètres des lois normales, il est possible de revenir à la section 2.1.5.1.2.

La Figure 2.9 montre que la régression double bi-exponentielle initiale est encore plus fidèlement suivie qu'auparavant.

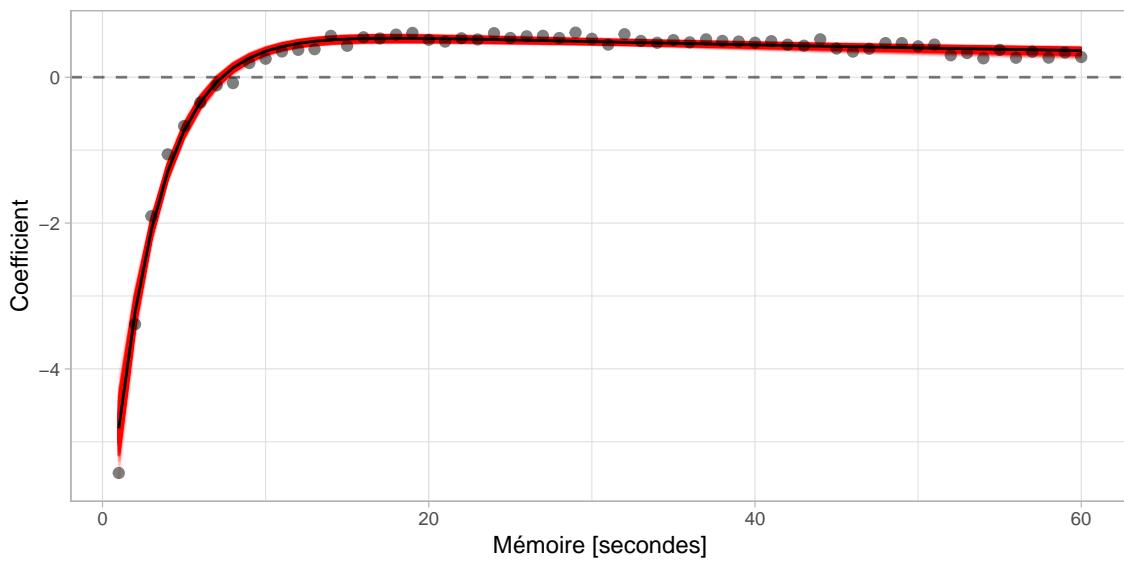


Figure 2.9: Variation des coefficients autorégressifs selon une loi normale multivariée.

Et surtout, comme nous pouvons le voir sur la Figure 2.10, il en va de même pour la profondeur, alors que précédemment les tirages donnaient des courbes fortement éloignées de celle attendue.

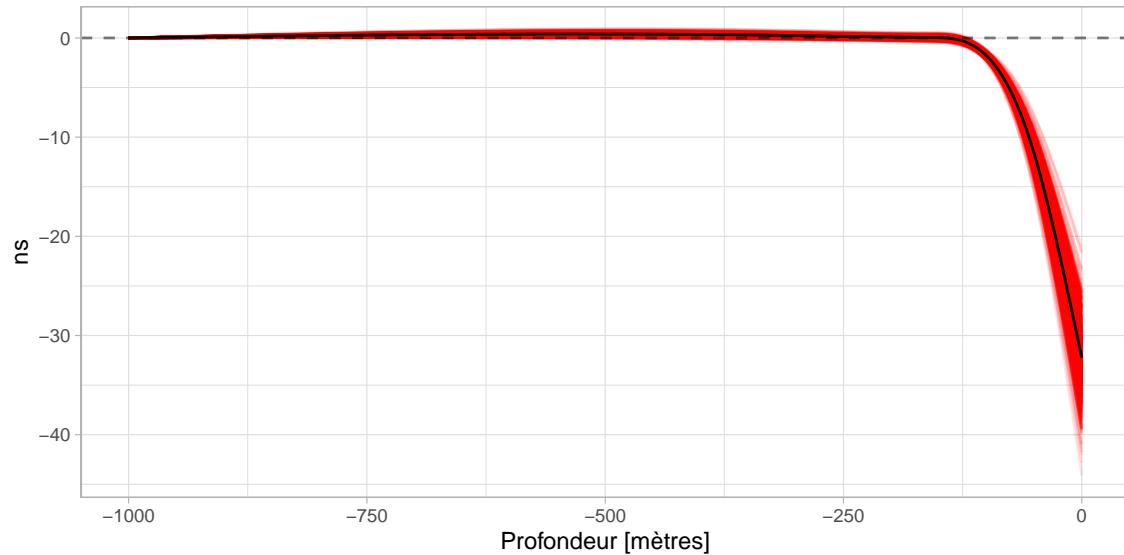


Figure 2.10: Variation des coefficients de profondeur selon une loi normale multivariée.

Sur la Table 2.4 nous pouvons constater que les intervalles de confiance estimés en utilisant des lois normales multivariées sont nettement plus petits et peuvent conduire à conclure sur un effet de l'exposition sur le taux d'émission de buzz.

Table 2.4: Intervalles de confiance dans le cas de normales multivariées.

	Monte-Carlo		Erreur standard	
	inf	sup	inf	sup
β_{E_0}	-4.888	-4.273	-4.632	-4.498
β_{E_1}	-1.971	-0.480	-1.237	-1.229
β_{E_2}	-64.615	-52.024	-58.310	-58.097
β_{E_3}	-124.493	-98.742	-111.628	-111.209

Nous pouvons également voir sur la Figure 2.11 que les distributions semblent normales, contrairement à celles observées avec les lois univariées.

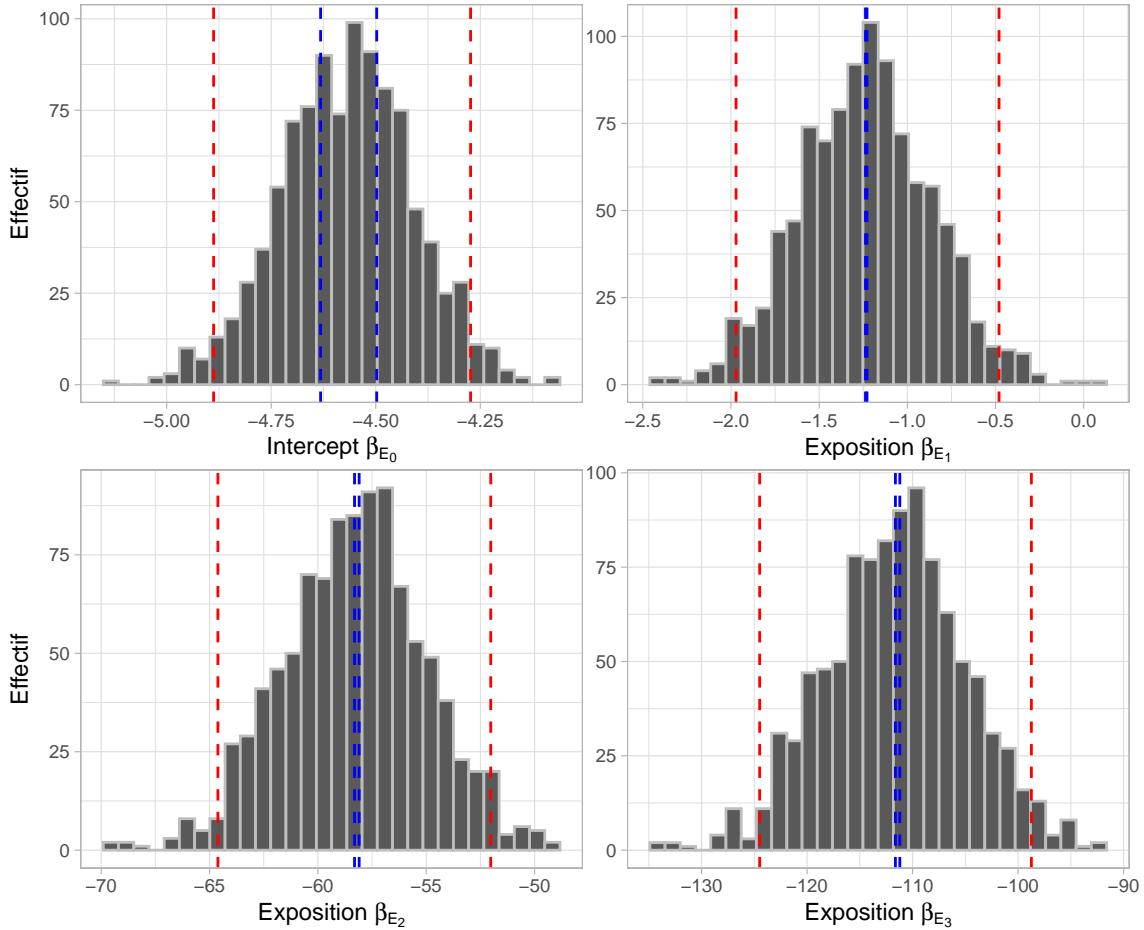


Figure 2.11: Intervalles de confiance dans le cas de normales multivariées.

2.2.5.1.3 Variance-covariance des coefficients autorégressifs Bien que plus intéressante, la procédure que nous avons mise en place capte la variabilité de la régression double bi-exponentielle et non celle du phénomène autorégressif en lui-même. Nous reprenons donc la procédure Monte-Carlo employée dans la section précédente, mais avec un vecteur de moyennes et une matrice de variance-covariance obtenus via une seconde procédure Monte-Carlo faisant varier les coefficients autorégressifs. L'ensemble de cette procédure est décrite par l'Algorithme 2 de la section 2.1.5.1.3.

Nous pouvons contrôler sur la Figure 2.12 que les régressions double bi-exponentielles obtenues restent cohérentes.

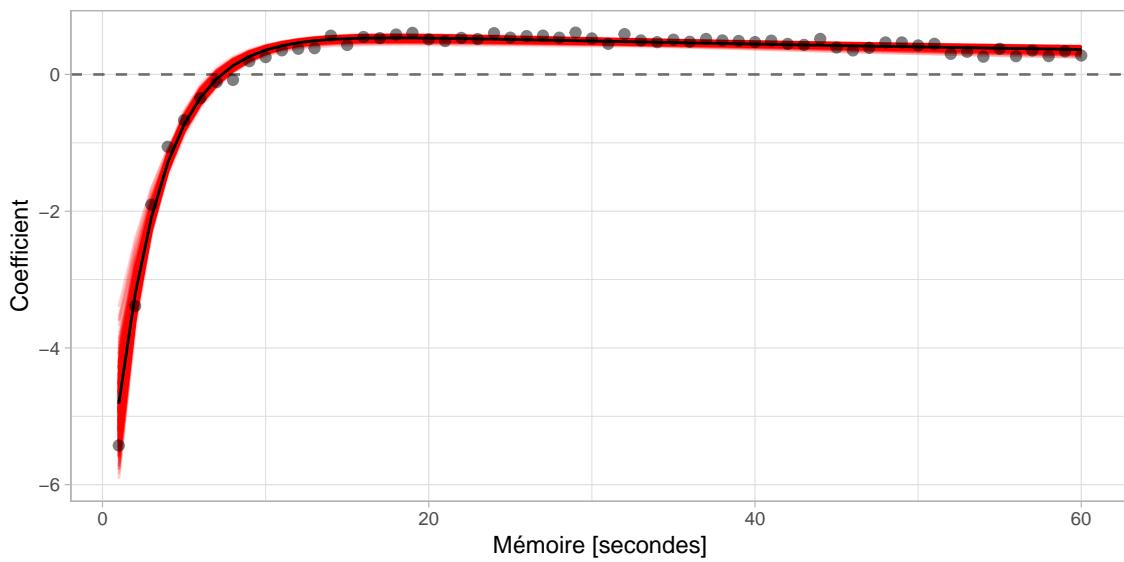


Figure 2.12: Variation des coefficients autorégressif selon une loi normale multivariée estimée par Monte-Carlo.

La Table 2.5 et la Figure 2.13 permettent de vérifier que bien que les intervalles calculés ainsi sont légèrement plus grands que les précédents, ils permettent toujours d'interpréter les coefficients d'exposition ajustés par le modèle.

Table 2.5: Intervalles de confiance dans le cas d'une normale multivariée estimée par Monte-Carlo.

	Monte-Carlo		Erreur standard	
	inf	sup	inf	sup
β_{E_0}	-4.866	-4.275	-4.645	-4.510
β_{E_1}	-1.981	-0.426	-1.122	-1.114
β_{E_2}	-65.140	-51.949	-59.343	-59.128
β_{E_3}	-125.572	-98.545	-113.695	-113.274

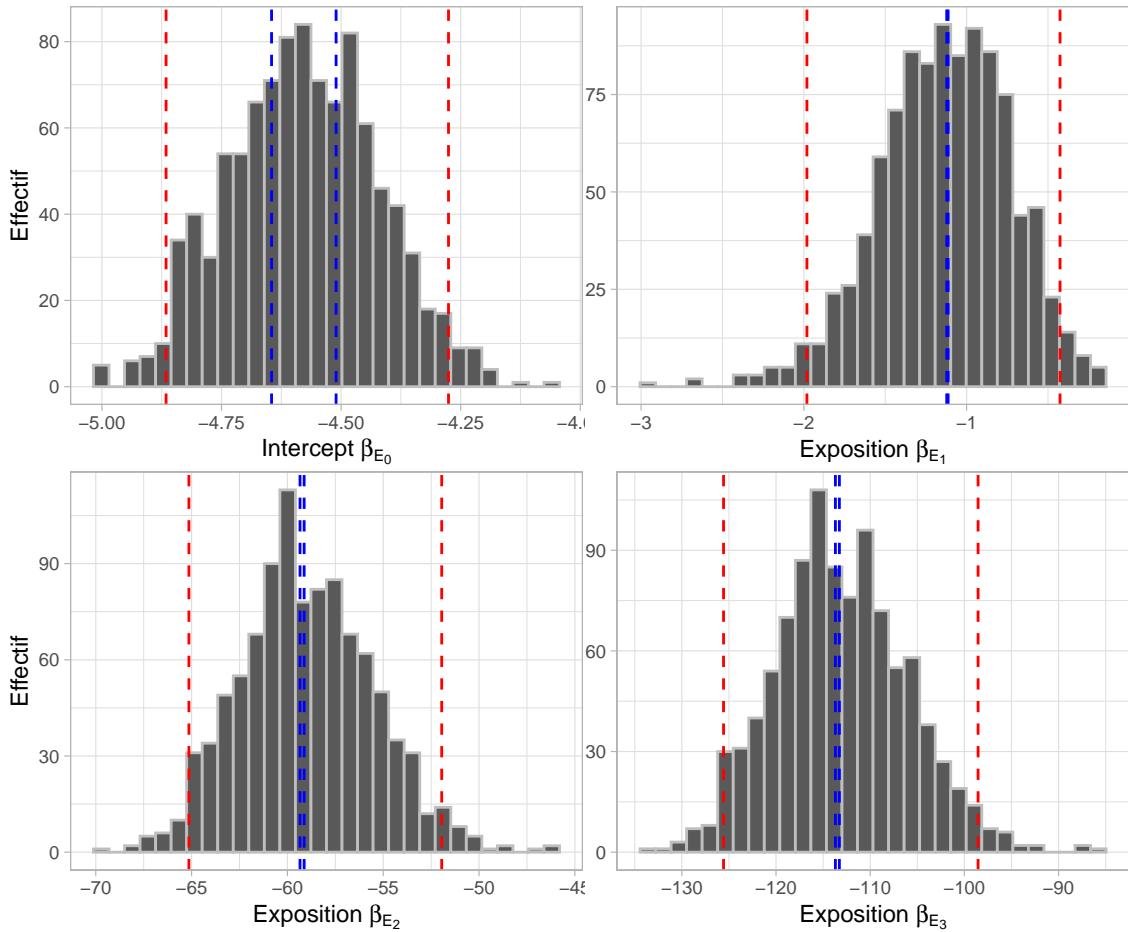


Figure 2.13: Intervalles de confiance dans le cas d'une normale multivariée estimée par Monte-Carlo.

2.2.5.1.4 Sans passer par la régression double bi-exponentielle Pour les raisons évoquées dans la section 2.1.5.1.4, il est également pertinent d'observer les intervalles de confiance des coefficients d'exposition en ayant calculé le terme d'offset directement à partir des coefficients autorégressifs, sans utiliser de régression double bi-exponentielle.

Nous pouvons voir sur la Table 2.6 et la Figure 2.14 qu'avec cette approche plus directe, les intervalles de confiance sont quasiment identiques à ceux obtenus dans la section précédente, et même plus petit pour l'ordonnée à l'origine.

Table 2.6: Intervalles de confiance dans le cas d'une loi normale multivariée sans double bi-exponentielle.

	Monte-Carlo		Erreur standard	
	inf	sup	inf	sup
β_{E_0}	-4.877	-4.261	-4.655	-4.513
β_{E_1}	-1.782	-0.566	-1.187	-1.178
β_{E_2}	-64.410	-52.093	-58.181	-57.969
β_{E_3}	-124.003	-99.056	-111.401	-110.984

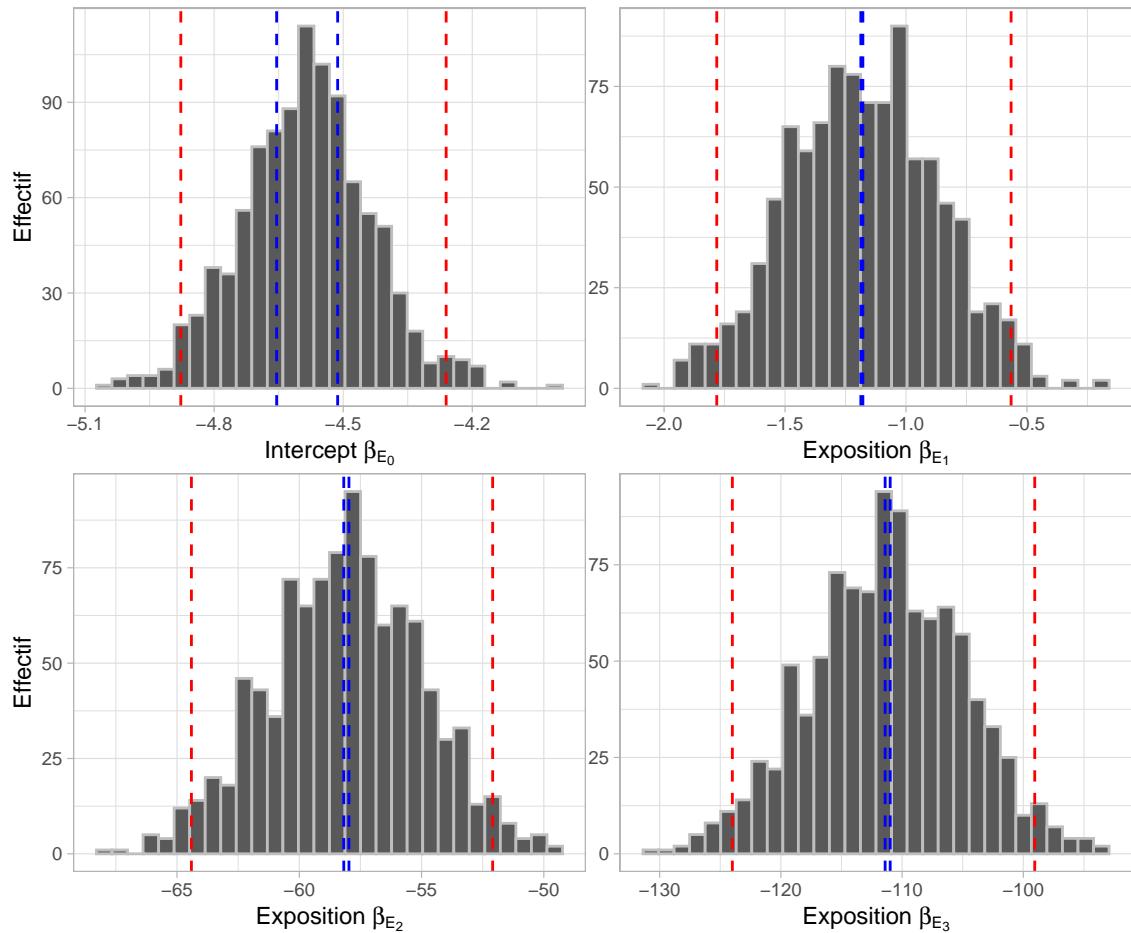


Figure 2.14: Intervalles de confiance dans le cas d'une loi normale multivariée sans double bi-exponentielle.

2.2.5.2 Pourcentage du taux normal d'émission de buzz

Il est rassurant de voir que les intervalles de confiance des coefficients d'exposition obtenus par Monte-Carlo restent raisonnablement petits, cependant comme évoqué précédemment, il n'est pas possible de lier une augmentation ou une diminution du taux d'émission de buzz en fonction de la distance avec le bateau à partir de leur valeur. C'est pourquoi nous avons représenté graphiquement cette évolution sur la Figure 2.4, mais cette visualisation ne proposait pas d'intervalle de confiance ou de bande de prédiction.

Nous avons employé la méthode Delta décrite dans la section 2.1.5.2 pour construire une bande de confiance autour de l'estimation du pourcentage du taux normal d'émission de buzz en fonction de la profondeur. Ce pourcentage est obtenu en faisant le rapport entre l'intensité estimée avec et sans perturbation. La Figure 2.15 permet de voir que la bande de confiance tracée en rouge est fine. Nous sommes donc confiants dans la lecture de la courbe qui illustre que soumis à des perturbations à une distance inférieure à 15 kilomètres, les narvals commencent à émettre nettement moins de buzz que dans des conditions normales. Afin de comparer l'approche de la méthode Delta, nous avons utilisé la fonction *predictInterval* du package **merTools**. Celle-ci permet d'obtenir une bande de prédiction pour des modèles mixtes par une approche Monte-Carlo. Comme nous pouvons le voir sur la Figure 2.15, la bande obtenue ainsi (tracée en bleue) suit la même forme que celle de la méthode Delta.

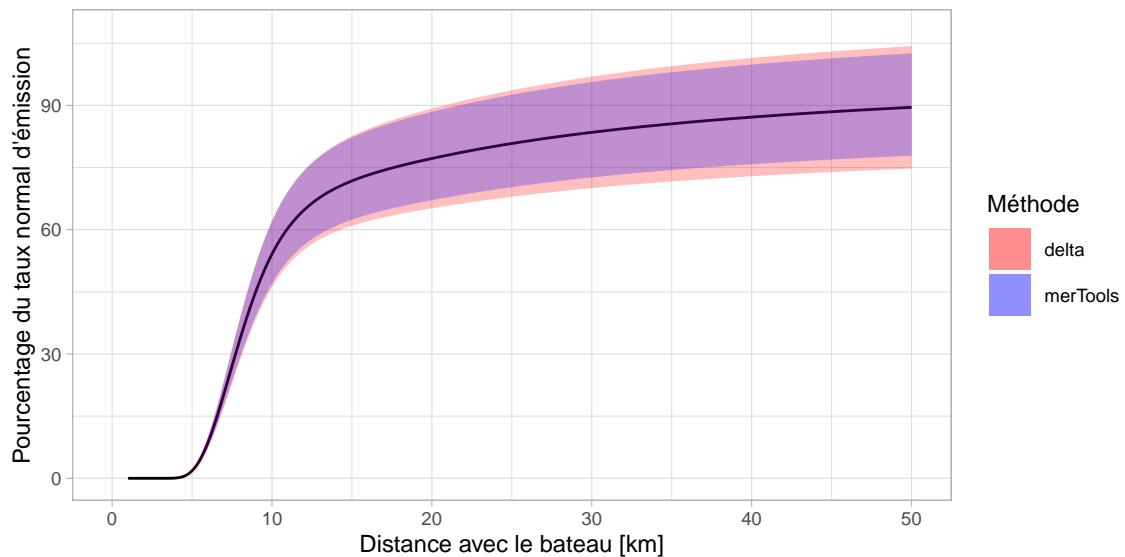


Figure 2.15: Pourcentage du taux normal d'émission de buzz selon la distance au bateau.

2.3 Conclusion

Les narvals sont des animaux très vocaux, ce qui leur permet notamment de trouver leurs proies et de se nourrir. Les scientifiques s'interrogent donc sur le potentiel impact de perturbations sonores liées aux activités humaines susceptibles de se développer au Groenland à cause du recul des glaces l'environnant.

Nous avons donc cherché à évaluer la différence des taux d'émission de buzz, les sons qu'utilisent les narvals pour chasser, entre des conditions "normales" et des conditions où les baleines sont exposées à des perturbations sonores. Pour ce faire nous avons utilisé les données de profondeur, de position et sonores collectées pendant plusieurs semaines en équipant 6 narvals d'instruments de mesure. Nous avons supposé que les buzz émis par les narvals suivent un processus de Poisson et nous nous sommes placés dans le cadre des modèles linéaires mixtes généralisés afin de modéliser le lien entre le taux d'émission des buzz, la profondeur de plongée des baleines et le niveau d'exposition aux perturbations.

Afin de quantifier la différence de taux d'émission entre des conditions "normales" et des conditions perturbées nous avons du tenir compte de l'effet de médiation de la profondeur. Pour cela nous avons estimé le lien entre le taux d'émission et la profondeur sans perturbation et nous avons injecté cette relation de référence dans le modèle incluant le niveau d'exposition au moyen d'un "offset". L'estimation des intervalles de confiance et de prédiction de l'effet des perturbations ne peut alors plus être faite en utilisant uniquement la variance des coefficients du modèle. C'est pourquoi nous avons mis en place une approche Monte-Carlo d'estimation des intervalles de confiance et que nous avons utilisé la méthode Delta pour construire des bandes de prédiction.

Il est apparu que les intervalles de confiance et les bandes de prédiction obtenues mettent en avant une réduction du taux d'émission de buzz par rapport à des conditions normales de plus en plus forte quand les narvals sont proches de la source des perturbations.

Cependant, plusieurs leviers peuvent être mis en place pour rendre plus fiable la modélisation du lien entre perturbations et émission de buzz :

- estimer l'effet direct, l'effet de médiation causale du triplet taux d'émission, profondeur et exposition aux perturbations,
- utiliser un type de processus autre que celui de Poisson, comme celui de Hawkes, potentiellement plus adapté.

Estimation de la durée de vie des narvals grâce aux sillons de leur dent

Chez les femelles narvals, les deux dents restent à l'intérieur de la boîte crânienne, tandis que pour les mâles, la canine gauche s'allonge et prend la forme d'une corne, comme le montre la Figure 3.1. Elle commence à pousser au travers de la lèvre supérieure gauche dès l'âge d'un an lors de la puberté et croît jusqu'à la maturité sexuelle, entre 8 et 9 ans. Cette défense torsadée possède des fonctionnalités et propriétés uniques dans la nature. Elle contient des millions de terminaisons nerveuses, ce qui en fait un organe sensoriel très développé [2].



Figure 3.1: Vue de face d'un narval et de sa dent. [3]

Certains chercheurs danois, comme Eva Garde, s'intéressent plus particulièrement à l'estimation de la durée de vie des narvals via l'information contenue dans cette dent. Pour mener cette étude, plusieurs découpes latérales des dents d'animaux décédés ont été réalisées. Comme nous pouvons le voir sur la Figure 3.2, ces découpes se présentent sous la forme d'une séquence de sillons ou de couches comportant des marqueurs saisonniers au cours de la croissance des dents. Ces derniers créent des motifs sinusoïdaux. La fréquence et la forme de ces sinusoïdes varient d'une année à l'autre selon la variabilité de la durée ou de l'intensité des saisons [4]. L'information portée par les motifs à l'intérieur des défenses est donc logiquement liée à la durée de vie de l'animal.



Figure 3.2: Présentation de l'allure d'une section en longueur d'une dent de narval. [4]

Le premier objectif pour cette problématique est le choix d'un modèle sinusoïdal pouvant représenter l'information contenue dans la dent de l'animal. À partir de cette forme de modèle et d'observations, le deuxième objectif sur lequel nous allons nous concentrer est celui de l'estimation des paramètres du modèle sinusoïdal. Nous présentons donc dans les parties suivantes, le modèle envisagé, notre démarche d'estimation de ces paramètres à partir d'un algorithme SAEM et les résultats obtenus sur des données simulées.

3.1 Modèle sinusoïdal

Comme nous l'avons évoqué précédemment, les motifs sinusoïdaux observés sont le reflet de la variabilité des saisons, ainsi ce motif n'est pas répété identiquement en fonction du temps. Ces variations complexifient donc la modélisation de cette information.

Les observations le long de la défense sont notées Y_i pour $i = 1, \dots, n$, avec la position correspondante sur la dent notée x_i . Le modèle est le suivant :

$$Y_i = f(x_i, \varphi) + \varepsilon_i$$

avec ε_i un bruit aléatoire suivant une loi normale de moyenne 0 et de variance ω^2 .

La fonction de régression $f(x, \varphi)$ est une fonction périodique sinusoïdale telle que :

$$f(x, \varphi) = A \sin(g(x) + b) + B \sin(2g(x) + 2b + \frac{\pi}{2})$$

avec

$$g(x) = ax + \xi_x$$

et finalement ξ_x , un processus aléatoire d'Ornstein-Uhlenbeck, tel que :

$$d\xi_x = -\beta \xi_x dx + \sigma dW_x$$

Dont la solution est donnée par :

$$\xi_{x+\Delta} = \xi_x e^{-\beta\Delta} + \int_x^{x+\Delta} \sigma e^{\beta(s-x)} dW_s$$

de sorte que la densité de transition est :

$$p(\xi_{x+\Delta} | \xi_x) = \mathcal{N}(\xi_x e^{-\beta\Delta}, \frac{\sigma^2}{2\beta}(1 - e^{-2\beta\Delta}))$$

où Δ est l'intervalle de temps entre deux observations.

Dans ce cadre, l'objectif est donc d'estimer les paramètres θ :

- $\varphi = (A, B, a, b)$,
- ω ,
- $\psi = e^{-\beta\Delta}$,
- $\gamma^2 = \frac{\sigma^2}{2\beta}(1 - \psi^2)$.

Une réalisation de ce modèle est présentée sur la Figure 3.3.

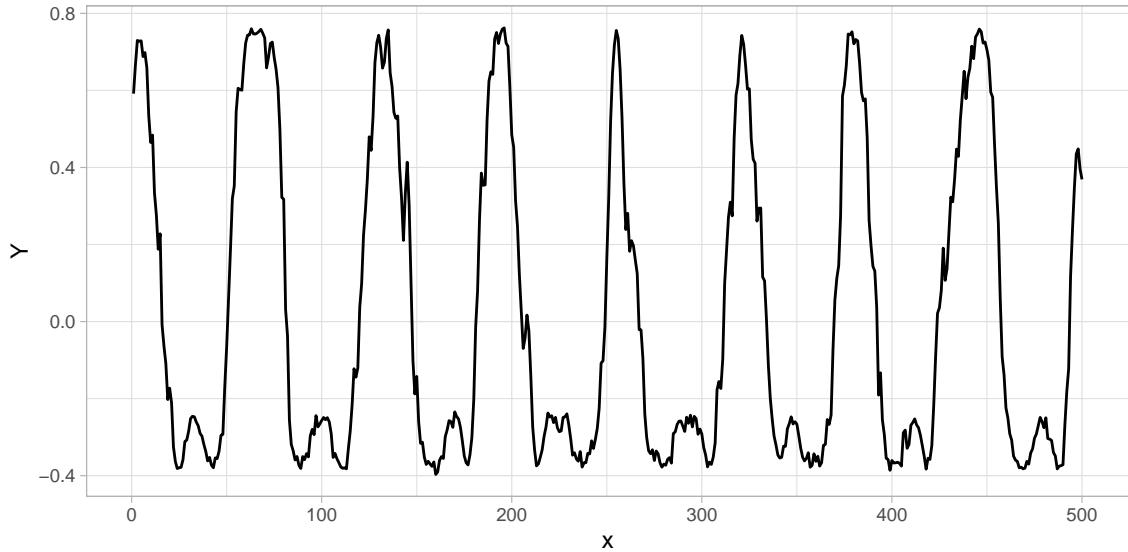


Figure 3.3: Simulation des observations Y , avec les paramètres suivants : $A = 0.5$, $B = -0.25$, $b = 1$, $a = 0.1$, $\beta = 0.05$, $\sigma = 0.1$, $\omega = 0.01$ et $\Delta = 1$.

3.1.1 Identifiabilité

Afin de justifier l'intérêt de l'estimation des paramètres du modèle, nous nous sommes intéressé à son identifiabilité. Nous pouvons observer sur la Figure 3.4 une trajectoire du processus ξ_x cible ainsi que trois autres simulations de trajectoire du processus ξ_x pour des valeurs de paramètres ψ et γ variant, obtenues selon le procédé suivant :

```
 $\xi_{1:n} \leftarrow 0$                                 ▷ Initialisation de la première valeur
 $\text{for } i \in \{1, \dots, n\} \text{ do}$ 
```

```

 $\xi_i = \xi_{i-1} * \psi + \epsilon_\xi$ , avec  $\epsilon_\xi \sim N(0, \gamma^2)$ 
end for

```

Nous observons sur la Figure 3.4 que les différentes réalisations du processus ξ_x conduisent à des trajectoires différentes. Comme attendu au regard de l'expression du processus, la modification de ψ impacte la valeurs moyenne de ξ_x , et γ sa variance.

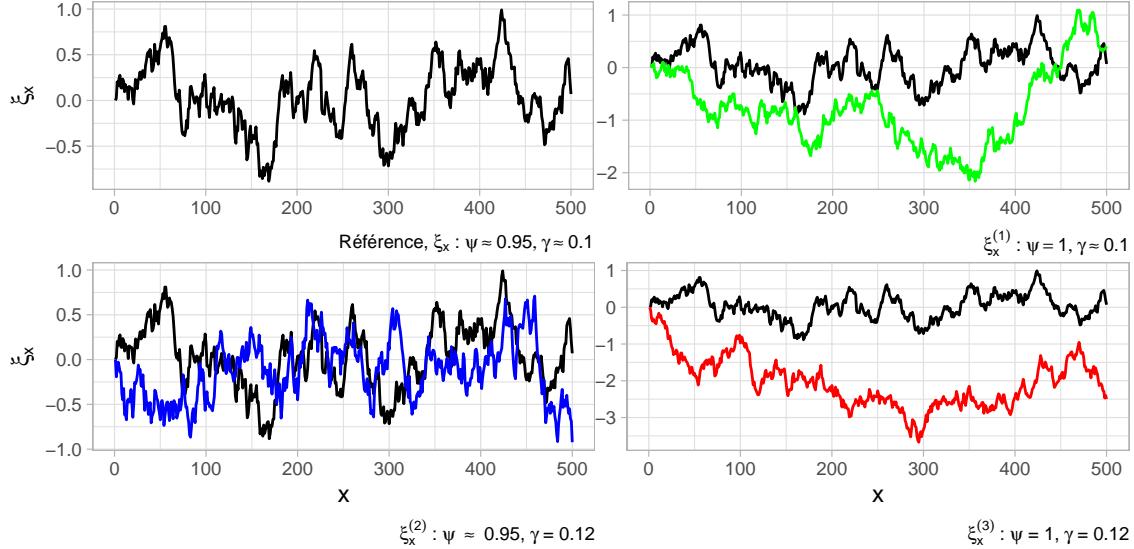


Figure 3.4: Présentation d'une trajectoire du processus ξ_x cible (en noir) et de trois autres trajectoires de ce processus simulées à partir de valeurs différentes de ψ et γ (en vert, bleu et rouge).

Ce résultat a un impact direct sur les observations Y puisque, comme nous pouvons le voir sur la Figure 3.5, les observations Y correspondant à des réalisations ξ_x pour des valeurs de ψ et γ différentes ne collent pas du tout à la distribution cible. De plus, une variation des paramètres A , B , a , et b entraîne une différence encore plus forte : A et B jouant sur l'amplitude de la sinusoïde, a sur sa pulsation et b sur son décalage de phase.

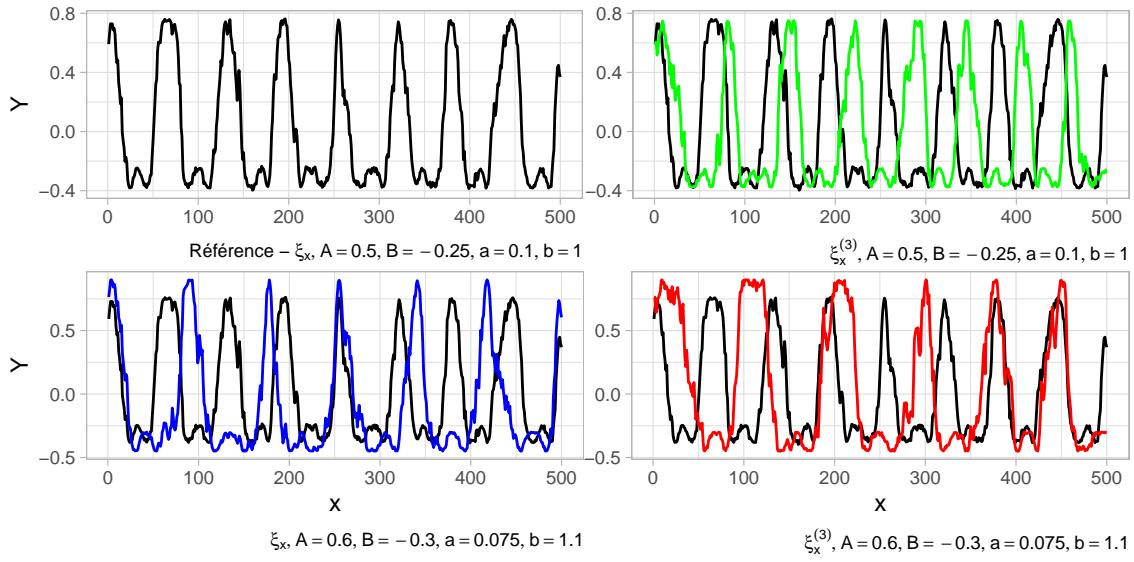


Figure 3.5: Distributions Y pour différentes trajectoires de ξ_x et différentes valeurs de A, B, a, b (en vert, bleu et rouge).

Les observations de Y sont sensibles à la trajectoire du processus ξ_x associé, ainsi qu'aux paramètres A, B, a, b . Le modèle sinusoïdal semble donc identifiable.

3.2 Estimation des paramètres à partir d'un algorithme SAEM

Afin d'estimer les paramètres θ du modèle présenté dans la partie précédente, nous avons implémenté une procédure reposant sur l'algorithme SAEM. Nous allons d'abord présenter le principe d'un algorithme EM (Espérance-Maximisation) [5], puis celui de son approximation stochastique : l'algorithme SAEM (Stochastic Approximation EM) [6]. Nous détaillerons les étapes MCMC [7] (Markov Chain Monte Carlo) et SMC [8] (Sequential Monte Carlo) avant de présenter l'algorithme complet.

3.2.1 Algorithme EM

L'algorithme EM est basé sur la log-vraisemblance complète du modèle qui s'écrit de la manière suivante :

$$\begin{aligned}
\log L(Y, \xi_x, \theta) &= \sum_{i=1}^n \log p(Y_i | \xi_i) + \sum_{i=1}^n \log p(\xi_i | \xi_{i-1}) + \log p(\xi_1) \\
&= - \sum_{i=1}^n \frac{(Y_i - f(x_i, \varphi))^2}{2\omega^2} - \frac{n}{2} \log(\omega^2) \\
&\quad - \sum_{i=1}^n \frac{(\xi_i - \xi_{i-1}\psi)^2}{\frac{\sigma^2}{\beta}(1-\psi^2)} - \frac{n}{2} \log\left(\frac{\sigma^2}{2\beta}(1-\psi^2)\right) \\
&= - \sum_{i=1}^n \frac{(Y_i - f(x_i, \varphi))^2}{2\omega^2} - \frac{n}{2} \log(\omega^2) \\
&\quad - \sum_{i=1}^n \frac{(\xi_i - \xi_{i-1}\psi)^2}{2\gamma^2} - \frac{n}{2} \log(\gamma^2)
\end{aligned}$$

Pour chaque itération k , l'algorithme EM procède aux deux étapes suivantes, étant donné la valeur courante des paramètres $\theta^{(k)}$.

- étape E : calcul de $Q(\theta, \theta^{(k)})$, l'espérance conditionnelle de la log-vraisemblance du modèle : $Q(\theta, \theta^{(k)}) \leftarrow E[\log L(Y, \xi, \theta) | Y; \theta^{(k)}]$
- étape M : actualisation des paramètres $\theta^{(k+1)} = \arg \max_{\theta} Q(\theta, \theta^{(k)})$.

Pour actualiser les paramètres, nous avons besoin de leurs statistiques exhaustives contenant toute l'information de la vraisemblance. En remarquant que la vraisemblance complète du modèle appartient à la famille exponentielle, nous obtenons les définitions suivantes :

$$\begin{aligned}
S_1(\xi_i) &= \frac{1}{n} \sum_{i=1}^n (Y_i - f(x_i(\xi_i), \varphi))^2 \\
S_2(\xi_i) &= \sum_{i=1}^n \xi_{i-1} \xi_i \\
S_3(\xi_i) &= \sum_{i=1}^n \xi_{i-1}^2 \\
S_4(\xi_i) &= \sum_{i=1}^n \xi_i^2
\end{aligned}$$

L'actualisation des paramètres dépend directement de ces statistiques.

3.2.2 Simulation de ξ_x

Dans notre cas, la distribution conditionnelle $p(\xi_x|Y; \theta^{(k)})$ n'est pas explicite en raison de la non-linéarité de notre fonction de régression $f(x, \varphi)$. Nous pouvons donc utiliser un algorithme MCMC ou un algorithme SMC pour simuler selon cette distribution.

3.2.2.1 Algorithme MCMC

L'objectif de cet algorithme MCMC est de simuler une trajectoire du processus ξ_x à partir des observations Y ainsi que des paramètres θ . L'algorithme programmé est plus précisément un algorithme de Gibbs - Metropolis Hasting avec marche aléatoire.

Effectivement, après l'initialisation d'une trajectoire $\xi^{(0)} = (\xi_1^{(0)}, \dots, \xi_n^{(0)})$, l'algorithme procède à M itérations. La trajectoire du processus simulée peut donc s'écrire $\xi^{(M)} = (\xi_1^{(M)}, \dots, \xi_n^{(M)})$.

Plus précisément, pour chaque itération k , on calcule pour chaque position x_i , une valeur courante candidate ξ_c avec une marche aléatoire : $\xi_i^{(c)} = \xi_i^{(k-1)} + N(0, \delta_i^2)$. Cela introduit un nouveau paramètre $\delta = (\delta_1, \dots, \delta_n)$ contrôlant la variance de la marche aléatoire. Pour chacun de ses candidats, une log-probabilité d'acceptation est calculée de la façon suivante :

$$\log(\alpha) = \min\left(\log\left(\frac{L(Y, \xi^{(c)})}{L(Y, \xi^{(k-1)})}\right), 1\right)$$

avec :

$$\begin{aligned} \log\left(\frac{L(Y, \xi^{(c)})}{L(Y, \xi^{(k-1)})}\right) &= \log(L(Y, \xi^{(c)})) - \log(L(Y, \xi^{(k-1)})) \\ &= -\frac{1}{2\omega^2} \sum_{i=1}^n (Y_i - f_\varphi(\xi^{(c)}))^2 - \frac{1}{2\frac{\gamma^2}{2}} \sum_{i=1}^n (\xi^{(c)} - \xi^{(k-1)}\psi)^2 \\ &\quad + \frac{1}{2\omega^2} \sum_{i=1}^n (Y_i - f_\varphi(\xi^{(k-1)}))^2 + \frac{1}{2\frac{\gamma^2}{2}} \sum_{i=1}^n (\xi^{(k-1)} - \xi^{(k-2)}\psi)^2 \end{aligned}$$

À partir de la valeur de cette log-probabilité ainsi que d'une réalisation d'une loi uniforme prenant ses valeurs entre 0 et 1, le candidat est soit rejeté, soit accepté, auquel cas, il remplace la valeur considérée à l'itération $k-1$.

De plus, nous avons choisi de rendre le paramètre δ adaptatif en fonction du taux d'acceptation acc_rate_i pour chaque point au fil des itérations k . Cela ajoute donc une étape d'actualisation à l'algorithme précédent, ce qui donne finalement l'Algorithme 3 :

Algorithme 3 Algorithme MCMC de simulation d'une trajectoire du processus ξ_x .

```

 $\xi_{1:n} \leftarrow 0$                                  $\triangleright$  Initialisation du processus
 $\delta_{1:n} \leftarrow 0.05$                              $\triangleright$  Initialisation du delta adaptatif
 $\delta_{AR} \leftarrow 0.1$                               $\triangleright$  Pas d'évolution du delta adaptatif
 $acc\_rate_{1:n} \leftarrow 0$                           $\triangleright$  Initialisation du vecteur de taux d'acceptation
 $acc\_rate_{target} \leftarrow 0.23$                     $\triangleright$  Taux d'acceptation visé
for  $k \in \{1, \dots, M\}$  do
    for  $i \in \{1, \dots, n\}$  do
         $\xi^{(c)} \leftarrow \xi$ 
         $\xi_i^{(c)} \sim \xi_i + \mathcal{N}(0, \delta_i^2)$            $\triangleright$  Simulation du candidat pour  $\xi_i$ 
         $\alpha_{log} \leftarrow \min(\log(\frac{L(Y, \xi^{(c)})}{L(Y, \xi)}), 1)$   $\triangleright$  Calcul de la probabilité d'acceptation
         $u \sim \mathcal{U}(0, 1)$                                 $\triangleright$  Tirage d'une réalisation de loi uniforme
        if  $\log u \leq \alpha_{log}$  then
             $\xi \leftarrow \xi^{(c)}$ 
        end if
         $acc\_rate_i \leftarrow$  mise à jour du taux d'acceptation
        if  $acc\_rate_i < acc\_rate_{target} * (1 - 0.1)$  then
             $\delta_i \leftarrow \delta_i * (1 - \delta_{AR})$             $\triangleright$  Réduction du delta adaptatif
        else if  $acc\_rate_i > acc\_rate_{target} * (1 + 0.1)$  then
             $\delta_i \leftarrow \delta_i * (1 + \delta_{AR})$             $\triangleright$  Augmentation du delta adaptatif
        end if
    end for
end for
 $\hat{\xi}_x \leftarrow \xi$ 

```

3.2.2.2 Algorithme SMC

Comme l'algorithme MCMC, l'algorithme SMC (ou filtre particulaire) a pour objectif de simuler une trajectoire de ξ_x à partir des observations Y et des paramètres θ .

Cependant son fonctionnement est assez différent : plutôt que d'accepter ou de rejeter un candidat pour chaque instant du processus selon un rapport de vraisemblance comme le fait l'approche MCMC, le SMC propose pour chaque instant P réalisations (les particules) selon les P estimations réalisées à l'instant précédent et associe à chacune des nouvelles réalisations un poids égale à la probabilité d'observer la valeur de Y à l'instant courant conditionnellement à la réalisation simulée. Il est alors possible de tirer avec remise parmi les particules en utilisant les poids normalisés comme probabilités de tirage et de conserver ainsi les particules permettant d'observer avec les plus fortes probabilités Y . Une fois que le dernier instant du processus est atteint, il suffit de tirer un index selon les derniers poids calculés pour obtenir une trajectoire du processus. L'Algorithme 4 détaille chacune de ces étapes :

Algorithme 4 Algorithme SMC pour la simulation d'une trajectoire du processus ξ_x .

```

 $w_{1:P} \leftarrow 1/P$                                  $\triangleright$  Initialisation des poids associés aux particules
 $\xi_c^{(1:P)} \leftarrow 0$                              $\triangleright$  Initialisation des particules au premier instant du processus
 $\xi_{1:n} \leftarrow \xi_c$                              $\triangleright$  Initialisation des trajectoires du processus
for  $i \in \{2, \dots, n\}$  do
    for  $j \in \{1, \dots, P\}$  do
         $\xi_c^{(j)} \sim \xi_{i-1}^{(j)} * \psi + \mathcal{N}(0, \gamma^2)$        $\triangleright$  Simulation d'une valeur courante selon  $\xi_{i-1}^{(j)}$ 
         $w_j \leftarrow p(Y_i | \xi_c^{(j)})$                            $\triangleright$  Poids égale à la probabilité de  $Y_i$  conditionnellement à  $\xi_c^{(j)}$ 
    end for
    for  $j \in \{1, \dots, P\}$  do
         $w_j \leftarrow \frac{w_j}{\sum_{k=1}^P w_k}$                        $\triangleright$  Normalisation
    end for
    for  $j \in \{1, \dots, P\}$  do
         $idx \leftarrow$  tirage probabiliste d'une particule en fonction des poids  $w$ 
         $\xi_i^{(j)} \leftarrow \xi_c^{(idx)}$                             $\triangleright$  Conservation de la particule  $idx$ 
    end for
end for
 $idx \leftarrow$  tirage probabiliste d'une trajectoire en fonction des poids  $w$ 
 $\hat{\xi}_x \leftarrow \xi^{(idx)}$ 

```

3.2.3 Algorithme SAEM

L'introduction d'une étape MCMC ou SMC conduit à la version stochastique de l'algorithme EM, à savoir l'algorithme SAEM. Cet algorithme utilise les étapes de l'algorithme EM auxquelles s'ajoute une étape d'approximation stochastique.

Effectivement, pour chaque itération k , les étapes sont les suivantes :

- étape E : simulation d'une nouvelle trajectoire de $\xi^{(k)}$ à l'aide d'un algorithme MCMC considérant $p(\xi|Y; \theta^{(k)})$ comme une distribution stationnaire,
- étape SA : approximation stochastique des statistiques exhaustives :

$$\begin{aligned}
 s_1^{(k)} &= s_1^{(k-1)} + \alpha_k(S_1(\xi^{(k)}) - s_1^{(k-1)}) \\
 s_2^{(k)} &= s_2^{(k-1)} + \alpha_k(S_2(\xi^{(k)}) - s_2^{(k-1)}) \\
 s_3^{(k)} &= s_3^{(k-1)} + \alpha_k(S_3(\xi^{(k)}) - s_3^{(k-1)}) \\
 s_4^{(k)} &= s_4^{(k-1)} + \alpha_k(S_4(\xi^{(k)}) - s_4^{(k-1)})
 \end{aligned}$$

- étape M : actualisation de θ^k , à partir des formules suivantes qui utilisent les statistiques exhaustives $s^{(k)}$:

$$\begin{aligned}
\hat{\varphi}^{(k)} &= \arg \min_{\varphi} \sum_{i=1}^n \left(Y_i - f(x_i(\xi_i^{(k)}), \varphi) \right)^2 \\
\widehat{\psi}^{(k)} &= \frac{s_2^{(k)}}{s_3^{(k)}} \\
\widehat{\omega^2}^{(k)} &= s_1^{(k)} \\
\widehat{\gamma^2}^{(k)} &= \frac{1}{n} (\widehat{\omega^2}^{(k)} s_3^{(k)} - 2\widehat{\psi}^{(k)} s_2^{(k)} + s_4^{(k)})
\end{aligned}$$

Au principe général d'un algorithme SAEM à Q itérations, nous avons ajouté deux hyper-paramètres M_{max} & α_{min} :

- Pour chaque itération q de l'algorithme SAEM, au moins une itération de l'algorithme MCMC ou SMC est effectuée. Dans le cas de l'utilisation d'une étape MCMC, afin de pouvoir améliorer les performances au début de notre algorithme, nous avons décidé de fixer ce nombre d'itérations à 5 pour les M_{max} premières itérations du SAEM, puis l'algorithme n'accomplit plus qu'une seule itération du MCMC. Le paramètre M_{max} est donc un hyper-paramètre de l'algorithme SAEM à choisir au préalable.
- Le deuxième paramètre concerne l'approximation stochastique effectuée pour chacune des itérations q . Effectivement, durant les premières itérations les approximations sont relativement éloignées de la valeur cible et donc sensiblement différentes entre elles. Au contraire, dans les dernières itérations, étant donné le phénomène de convergence que nous devons observer, les approximations sont censées être plus proches de la valeur cible et également entre elles. Afin de prendre en compte ce phénomène, nous faisons varier la valeur du paramètre de mémoire α permettant de tenir compte des valeurs précédemment estimées au fil des itérations à partir du paramètre α_{min} de la façon suivante :
 - dans un premier temps $\alpha = 1$ pour les α_{min} premières itérations, ce qui permet, d'appliquer la formule complète d'approximation,
 - puis, pour les $(Q - \alpha_{min})$ dernières itérations, les alphas sont calculés de la façon suivante :

$$\alpha_{\alpha_{min}:Q} = \frac{1}{l^{0.8}}, \text{ avec } l = 1 : (Q - \alpha_{min})$$

Ainsi le paramètre α décroît au fil des itérations à partir de la α_{min} ^{ième} itération. Cela permet de réduire l'importance du terme associé à α et d'augmenter donc l'influence de la valeur précédente pour ces itérations-là. Le choix de la puissance au dénominateur dans l'expression des $\alpha_{\alpha_{min}:Q}$ n'est pas complètement anodin. Les résultats théoriques sur la convergence de l'algorithme SAEM vers un optimum local reposent en partie sur les 2 hypothèses suivantes :

$$\sum_{i=1}^{\infty} \alpha_i = \sum_{n=1}^{\infty} \frac{1}{n^p} = \infty$$

$$\sum_{n=1}^{\infty} \alpha_n^2 = \sum_{n=1}^{\infty} \frac{1}{n^{2p}} < \infty$$

Cela implique $0.5 < p \leq 1$, d'où notre choix de $p = 0.8$.

L'Algorithme 5 résume l'ensemble de cette procédure SAEM :

Algorithme 5 Algorithme SAEM complet

```

 $s_1 \leftarrow (s_1^{(1)}, \dots, s_1^{(Q)})$ 
 $s_2 \leftarrow (s_2^{(1)}, \dots, s_2^{(Q)})$ 
 $s_3 \leftarrow (s_3^{(1)}, \dots, s_3^{(Q)})$ 
 $s_4 \leftarrow (s_4^{(1)}, \dots, s_4^{(Q)})$ 
 $\hat{\varphi} \leftarrow (\hat{\varphi}^{(1)}, \dots, \hat{\varphi}^{(Q)})$ 
 $\hat{\psi} \leftarrow (\hat{\psi}^{(1)}, \dots, \hat{\psi}^{(Q)})$ 
 $\hat{\omega}^2 \leftarrow (\hat{\omega}^{2(1)}, \dots, \hat{\omega}^{2(Q)})$ 
 $\hat{\gamma}^2 \leftarrow (\hat{\gamma}^{2(1)}, \dots, \hat{\gamma}^{2(Q)})$ 
 $\alpha_{1:\alpha_{min}-1} \leftarrow 1 ; \alpha_{\alpha_{min}:Q} \leftarrow \frac{1}{l^{0.8}}$  avec  $l = 1 : (Q - \alpha_{min} + 1)$            ▷ Initialisation du paramètre mémoire
 $M_{1:M_{max}} \leftarrow 5 ; M_{M_{max}+1:Q} \leftarrow 1$            ▷ Initialisation du nombre d'itérations du MCMC
for  $q \in \{2, \dots, Q\}$  do                                ▷ Etape E
     $\xi^{(q)} \leftarrow \xi$  avec  $\theta^{(q-1)}$ , par  $M_q$  itérations MCMC (Algorithme 3), ou par SCM (Algorithme 4)          ▷ Etape SA
     $S_1 \leftarrow \frac{1}{n} \sum_{i=1}^n (Y_i - f(x_i(\xi_i^{(q)}), \varphi))^2$            ▷ Calcul des statistiques exhaustives
     $S_2 \leftarrow \sum_{i=1}^n \xi_{i-1}^{(q)} \xi_i^{(q)}$ 
     $S_3 \leftarrow \sum_{i=1}^n (\xi_{i-1}^{(q)})^2$ 
     $S_4 \leftarrow \sum_{i=1}^n (\xi_i^{(q)})^2$            ▷ Mise à jour des approximations stochastiques
     $s_1^{(q)} \leftarrow s_1^{(q-1)} + \alpha_q (S_1(\xi^{(q)}) - s_1^{(q-1)})$ 
     $s_2^{(q)} \leftarrow s_2^{(q-1)} + \alpha_q (S_2(\xi^{(q)}) - s_2^{(q-1)})$ 
     $s_3^{(q)} \leftarrow s_3^{(q-1)} + \alpha_q (S_3(\xi^{(q)}) - s_3^{(q-1)})$ 
     $s_4^{(q)} \leftarrow s_4^{(q-1)} + \alpha_q (S_4(\xi^{(q)}) - s_4^{(q-1)})$            ▷ Etape M
     $\hat{\varphi}^{(q)} \leftarrow \arg \min_{\varphi} \sum_{i=1}^n \left( Y_i - f(x_i(\xi_i^{(q)}), \varphi) \right)^2$            ▷ Actualisation de  $\theta^{(q)}$ 
     $\hat{\psi}^{(q)} \leftarrow \frac{s_2^{(q)}}{s_3^{(q)}}$ 
     $\hat{\omega}^{2(q)} \leftarrow s_1^{(q)}$ 
     $\hat{\gamma}^{2(q)} \leftarrow \frac{1}{n} (\hat{\omega}^{2(q)} s_3^{(q)} - 2\hat{\psi}^{(q)} s_2^{(q)} + s_4^{(q)})$ 
end for

```

A chaque itération de l'algorithme, les paramètres $A, B, a, b = \varphi$ sont estimés en minimisant

les moindres carrés non linéaires. Sous **R**, cela se fait au moyen de la fonction *nls* qui nécessite des valeurs initiales des paramètres à estimer. Afin de favoriser la convergence de l'algorithme de Gauss-Newton utilisé pour la résolution du problème des moindres carrées non linéaires, nous avons tenu compte de l'influence des paramètres A et B sur l'amplitude de Y en les initialisant selon une réalisation de loi uniforme $\mathcal{U}(-\max(|Y|) - \omega, \max(|Y|) + \omega)$. Dans le même objectif, puisque le paramètre a est lié à la pulsation de la sinusoïde, il est initialisé selon $\hat{a} = 2 * \pi * f_{max}$ où f_{max} est la fréquence de Y avec la plus grande densité spectrale. Ce choix d'initialisation s'est avéré indispensable pour résoudre le problème des moindres carrées non linéaires. Le paramètre b influe lui sur le décalage de phase de la sinusoïde, nous pouvons donc aider l'algorithme en l'initialisant par $\hat{b} = (7 * \pi / 8) - x_0 * \hat{a}$ où x_0 correspond au plus petit x_i pour lequel Y est nul et \hat{a} à la définition précédente.

Par ailleurs, nous avons fait le choix d'initialiser $\hat{\omega}, \hat{\psi}, \hat{\gamma}$ à 0.5 pour la première itération.

3.3 Simulation & résultats

Dans cette dernière partie, nous présentons les résultats d'estimation de ξ_x par les algorithmes MCMC et SMC ; et plus largement de l'ensemble de coefficients θ par l'algorithme SAEM, ainsi que du plan d'expérience mis en place pour les valider.

3.3.1 Estimation de ξ_x par MCMC

Nous avons, dans un premier temps, testé l'efficacité de l'algorithme MCMC programmé indépendamment de l'algorithme SAEM. Pour ce faire, nous avons fixé les valeurs des paramètres : $A = 0.5$, $B = -0.25$, $b = 1$, $a = 0.1$, $\beta = 0.05$, $\sigma = 0.1$, $\omega = 0.01$ et $\Delta = 1$. Ces valeurs ont été utilisées pour simuler une trajectoire de ξ_x cible avec une distribution Y associée selon le modèle sinusoïdal. L'algorithme MCMC a été réalisé avec $M = 150$ itérations et obtient, à partir de ces mêmes valeurs de paramètres et des observations Y , les résultats présentés sur la Figure 3.6.

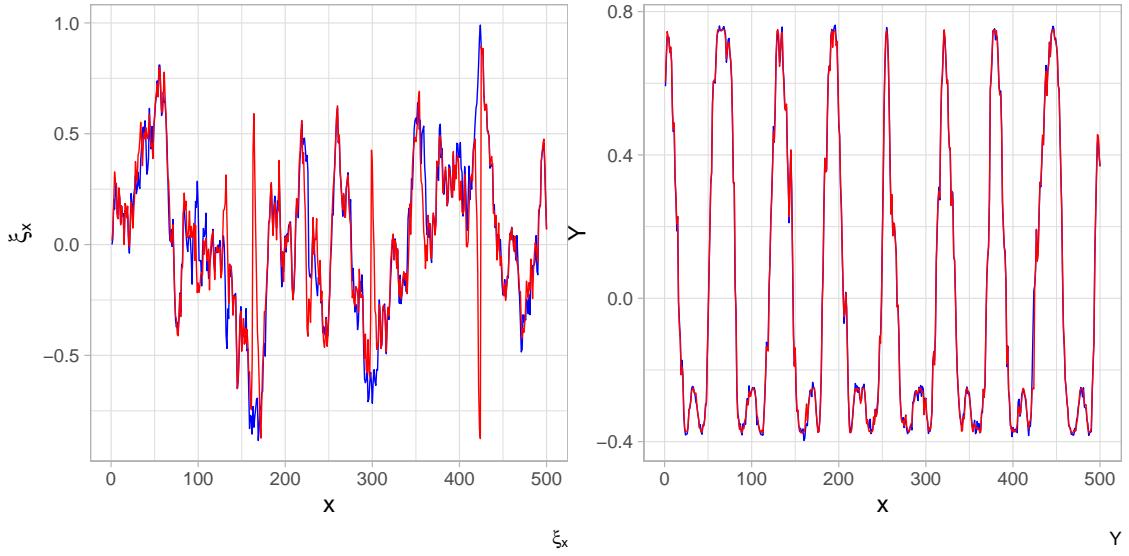


Figure 3.6: Superposition de la distribution Y cible et de la trajectoire ξ_x réellement utilisée (en bleu) ainsi que de la trajectoire de $\hat{\xi}_x$ obtenue par l'algorithme MCMC et la distribution \hat{Y} qui l'utilise (en rouge).

La trajectoire de ξ_x obtenue par le MCMC (en rouge) est très proche de celle à l'origine des observations Y . Nous remarquons cependant que quelques points de ξ_x sont très mal approchés par l'algorithme. Effectivement pour plusieurs positions x_i , la distance entre la valeur originelle de ξ_i et la valeur estimée par l'algorithme semble grande. Cependant, ces différences n'influencent que très peu l'allure du signal Y , qui reste très satisfaisant. Afin de comprendre la raison de ces erreurs, nous nous sommes intéressés au comportement du δ_i au fil des itérations, relativement à celui du taux d'acceptation acc_rate_i pour un des points concernés. Nous avons comparé leur évolution pour la position x_i où l'erreur est maximale et celle où elle est minimale.

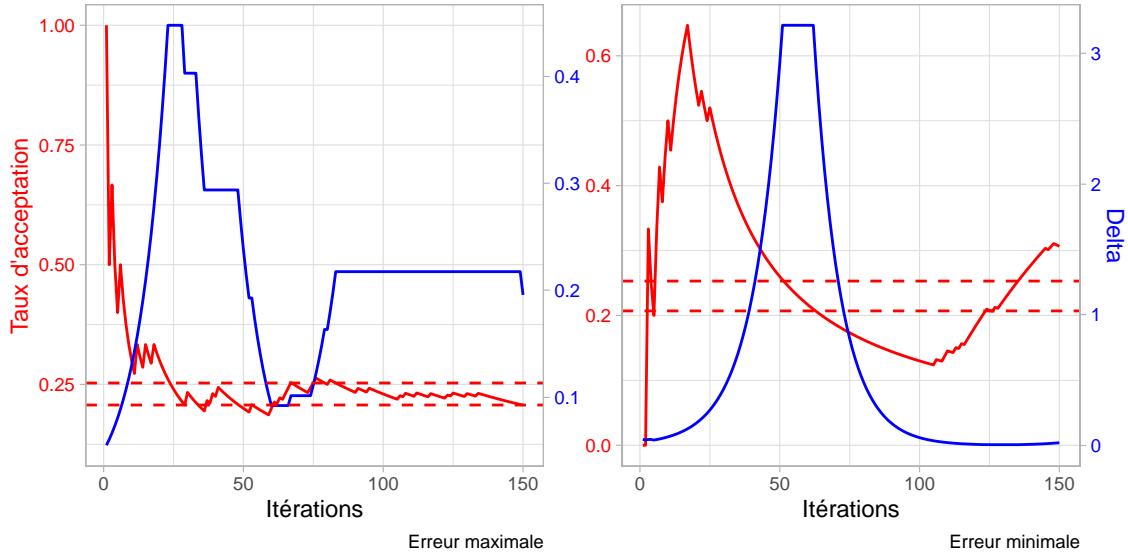


Figure 3.7: Distributions du taux d’acceptation et du delta adaptatif au fil des itérations pour deux positions différentes : celle pour laquelle l’erreur d’estimation est maximale et celle où elle est minimale.

Nous pouvons voir sur la Figure 3.7, que dans les deux cas, le delta adaptif évolue correctement. Effectivement le taux d’acceptation est premièrement supérieur à la borne supérieure de notre seuil ce qui entraîne l’augmentation de δ . Cette augmentation permet alors au taux de diminuer et donc de rentrer dans nos deux bornes de seuil, et δ stagne durant cette période. Le taux d’acceptation devient ensuite trop bas, et alors la valeur de δ diminue afin de le faire augmenter à nouveau. Nous remarquons seulement que, dans le cas concernant l’erreur maximale, l’évolution du taux d’acceptation est moins rapide que dans le cas de l’erreur minimale, par exemple il atteint l’intervalle de seuil environ vers la 40^{ème} itération contre environ la 20^{ème} pour le point où l’erreur est minimale. Cela entraîne une valeur maximale de δ bien plus élevée, d’environ 3 contre 0.4. Malgré cette observation, nous n’identifions pas la source de ces points aberrants. Cependant, étant donné que ceux-ci n’impactent pas l’estimation de la distribution des observations Y , l’algorithme MCMC programmé reste selon nous satisfaisant.

3.3.2 Estimation de ξ_x par SMC

Nous avons repris la trajectoire ξ_x à estimer dans la partie précédente et nous cette fois-ci tenté de la générer par notre algorithme SMC en utilisant $P = 500$ particules.

La Figure 3.8 permet de constater que cette approche semble plus précise que la MCMC : les points aberrants ont disparu et l’allure de ξ_x est bien reconstruite.

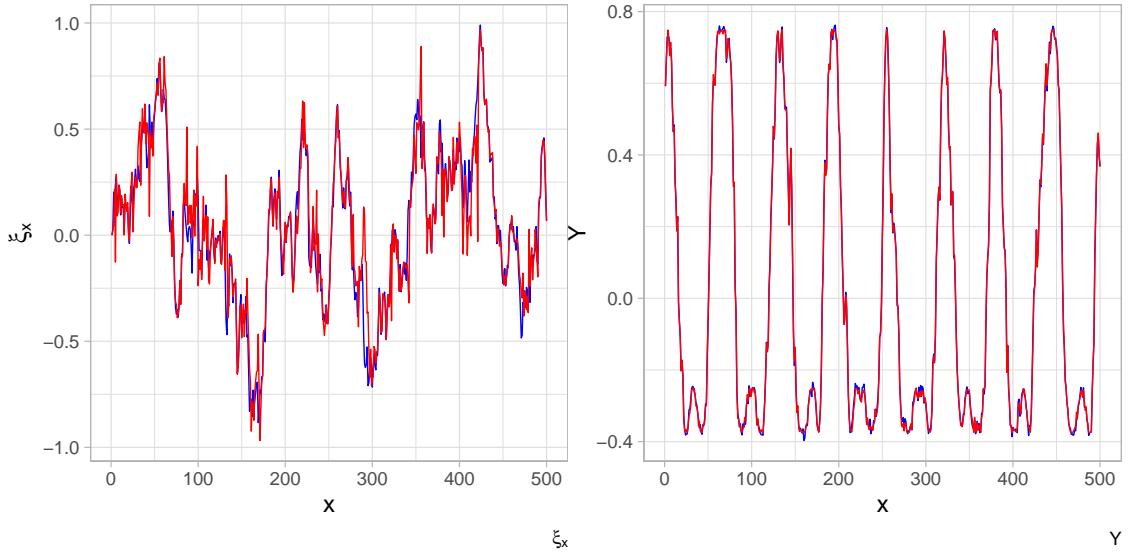


Figure 3.8: Superposition de la distribution Y cible et de la trajectoire ξ_x réellement utilisée (en bleu) ainsi que de la trajectoire de $\hat{\xi}_x$ obtenue par l'algorithme SMC et la distribution \hat{Y} qui l'utilise (en rouge).

3.3.3 Estimation de θ par SAEM

La validation de l'échantillonnage de ξ_x par MCMC ou SMC n'était qu'une étape, l'objectif global étant l'estimation de l'ensemble des paramètres θ . Nous avons donc ensuite observé le comportement de l'algorithme SAEM présenté précédemment. Nous présentons dans un premier temps les résultats obtenus pour la reconstruction d'un échantillon simulé avec une étape MCMC ou une étape SMC, puis la synthèse d'un plan d'expérience réalisé avec 1000 échantillons simulés.

3.3.3.1 Avec une étape MCMC

Pour la version SAEM-MCMC nous avons réalisé $Q = 500$ itérations de l'algorithme SAEM et après plusieurs tests et analyses graphiques, α_{min} a été fixé à 90 et M_{max} à 20, ce qui correspond à des ordres de grandeur régulièrement employés dans la littérature. La trajectoire de $\hat{\xi}_x$ obtenue après les 500 itérations de SAEM ainsi que les observations \hat{Y} correspondantes sont présentées sur la Figure 3.9. Nous observons sur cette figure que la reconstruction du processus ξ_x initial est un peu moins bonne que celle réalisée en connaissant les paramètres θ du modèle sinusoïdal, ce à quoi nous nous attendions. L'estimation de ξ_x demeure cependant très correcte et, conjointement à l'estimation des paramètres θ , permet une reconstruction fidèle de Y .

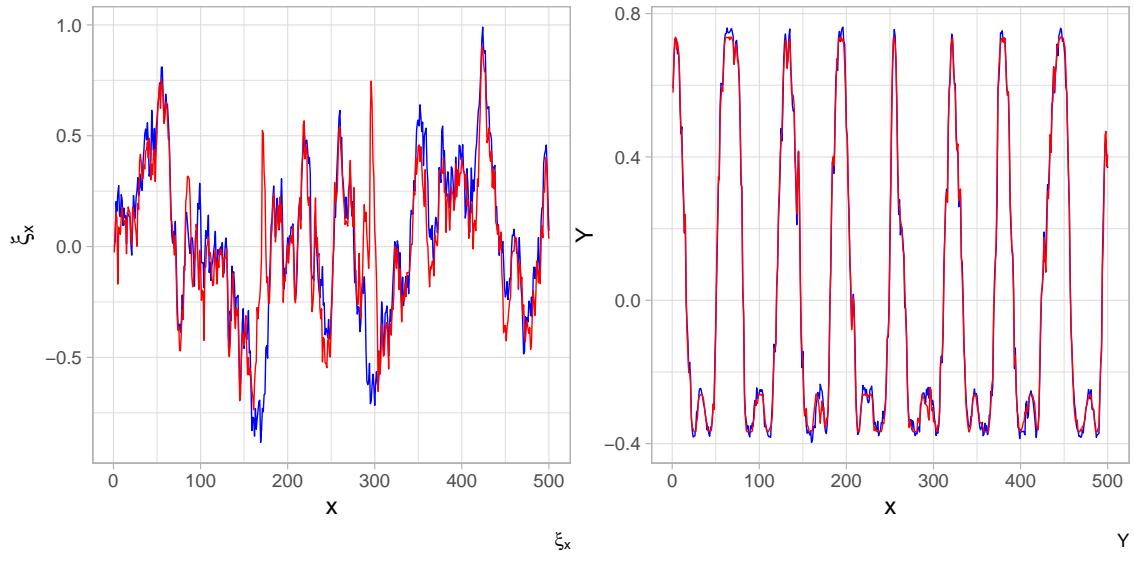


Figure 3.9: Superposition de la distribution Y cible et de la trajectoire ξ_x réellement utilisée (en bleu) ainsi que de la trajectoire de ξ_x obtenue par l'algorithme SAEM et la distribution Y qui l'utilise (en rouge).

La Figure 3.10 présente justement les valeurs estimées au fil des itérations pour ω, ψ, γ . Comme nous pouvons le voir sur cette figure, les paramètres ω et ψ convergent correctement vers les vraies valeurs des paramètres. Cependant, celle du paramètre γ ne converge pas exactement vers la valeur attendue et est légèrement surestimée.

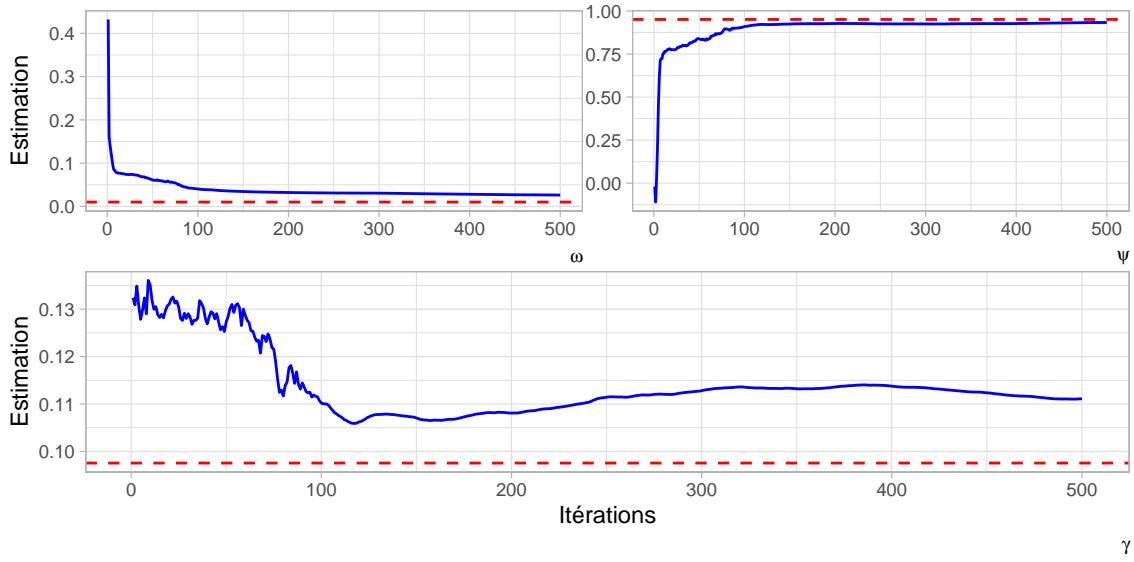


Figure 3.10: Présentation de l'estimation des coefficients ω , ψ et γ par l'algorithme SAEM, ainsi que des droites représentant la valeur cible de ces paramètres (en rouge).

Comme nous pouvons le voir sur la Figure 3.11, l'estimation du coefficient A converge assez normalement vers 0.5. Bien qu'elle se rapproche de -0.25 , celle de B semble avoir du mal à converger réellement et stagne au-dessus de ce que nous souhaiterions. Par ailleurs, nous observons, que l'estimation du paramètre a est très proche de la “vraie” valeur, mais nous ne voyons pas de convergence à proprement parler. Pour finir, l'estimation de b semble plus difficile à accomplir : nous ne remarquons pas de convergence vers la valeur attendue et l'estimation finale apparaît un peu éloignée de la valeur fixée initialement.

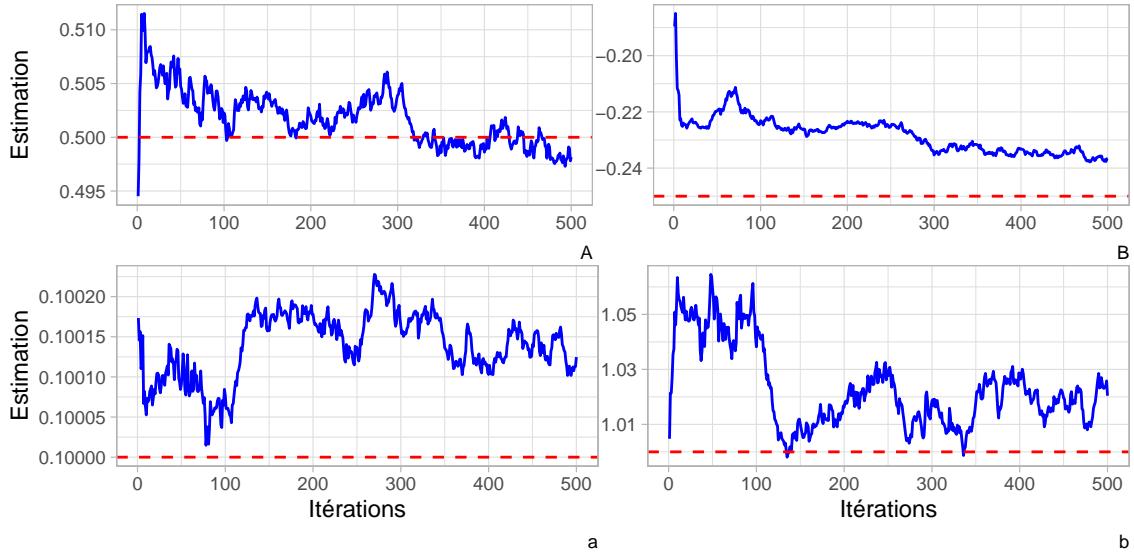


Figure 3.11: Présentation de l'estimation des coefficients A , B , a et b par l'algorithme SAEM, ainsi que des droites représentant la valeur cible de ces paramètres (en rouge).

3.3.3.2 Avec une étape SMC

Comme nous pouvons le voir sur les Figures 3.13 et 3.14, l'utilisation d'une étape SMC accélère sensiblement la convergence de $\hat{\theta}$, aussi nous avons utilisé seulement $Q = 100$ itérations de l'algorithme SAEM et nous avons diminué α_{min} à 25. L'étape SMC implique $P = 500$ particules.

La Figure 3.12 montre que cette version de SAEM-SMC permet une reconstruction de ξ_x légèrement meilleure que la version SAEM-MCMC, comme lorsque nous estimions $\hat{\xi}_x$ en fixant les paramètres θ . Cela conduit à une reconstruction finale \hat{Y} également très proche de Y .

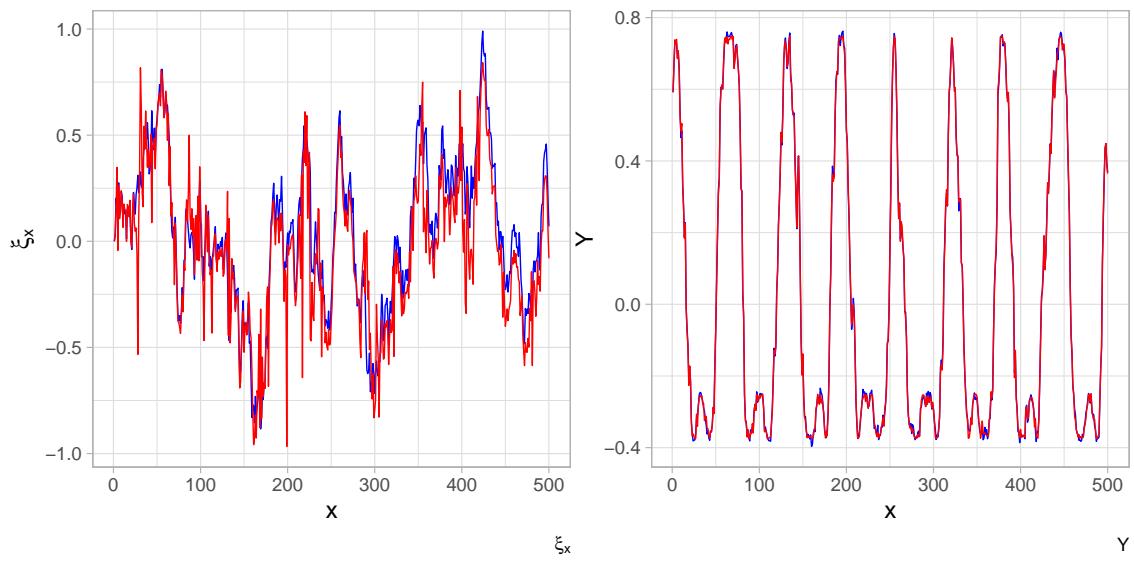


Figure 3.12: Superposition de la distribution Y cible et de la trajectoire ξ_x réellement utilisée (en bleu) ainsi que de la trajectoire de ξ_x obtenue par l'algorithme SAEM et la distribution Y qui l'utilise (en rouge).

En revanche la Figure 3.13 suggère que les estimations $\hat{\psi}$ et $\hat{\gamma}$ sont moins bonnes que précédemment : la convergence se fait vers des valeurs respectivement inférieure et supérieure à celles attendues. $\hat{\omega}$ semble elle mieux estimée.

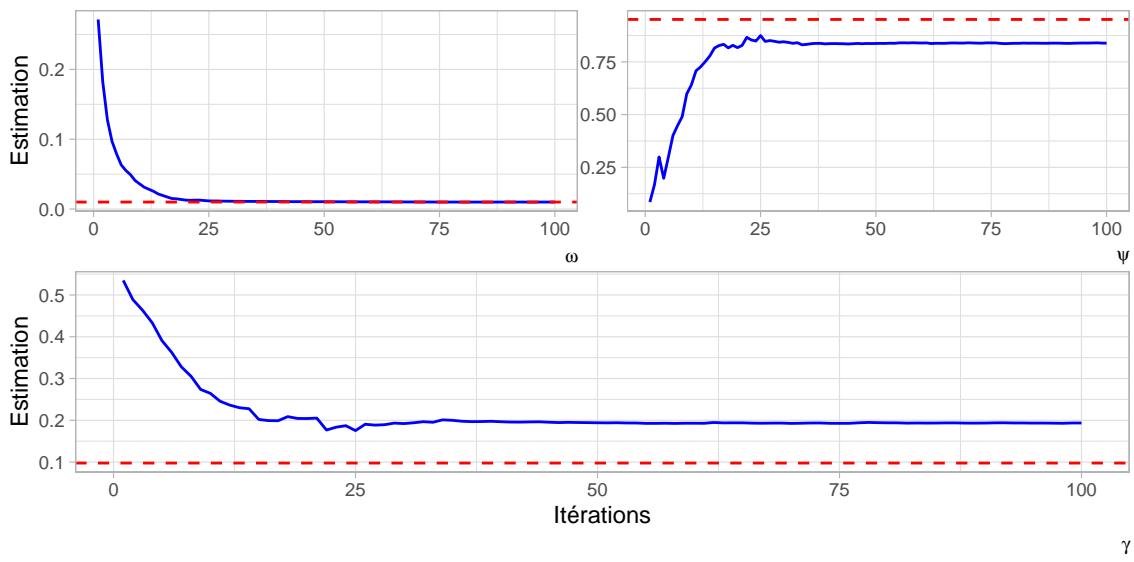


Figure 3.13: Présentation de l'estimation des coefficients ω , ψ et γ par l'algorithme SAEM, ainsi que des droites représentant la valeur cible de ces paramètres (en rouge).

Enfin, la Figure 3.14 indique que les paramètres $\hat{A}, \hat{B}, \hat{b}$ convergent mieux vers les valeurs fixées des paramètres avec la version SMC de l'algorithme. Le paramètre \hat{a} ne converge toujours pas réellement et est un peu plus éloigné qu'avec l'utilisation d'une étape MCMC, mais l'estimation reste très bonne.

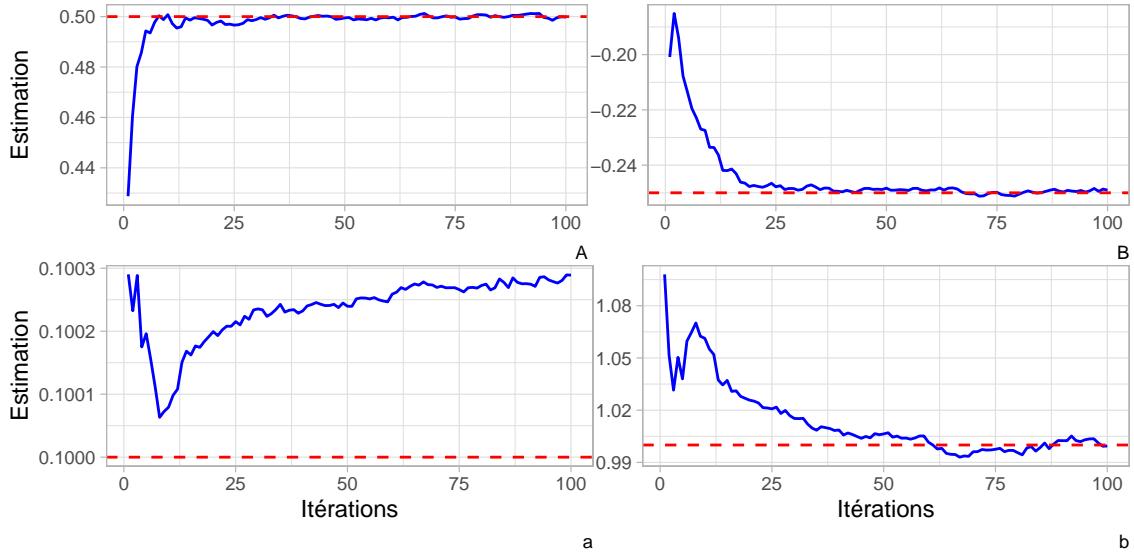


Figure 3.14: Présentation de l'estimation des coefficients A , B , a et b par l'algorithme SAEM, ainsi que des droites représentant la valeur cible de ces paramètres (en rouge).

3.3.3.3 Plan d'expérience

Les estimations de θ et ξ_x comportent une part d'aléatoire, ainsi afin d'évaluer les performances de notre algorithme SAEM il convient de répéter un grand nombre de fois ces estimations à partir de réalisations de ξ_x différentes. En effet, notre objectif est d'estimer nos paramètres sur un grand nombre d'itérations, afin d'analyser leurs variations autour des valeurs fixées. Nous avons choisi d'effectuer $M = 1000$ estimations de chacun des paramètres $\gamma, \psi, \omega, A, B, a, b$, et de calculer la racine carrée de l'erreur quadratique moyenne ($RMSE$), l'erreur en pourcentage (PE) et sa moyenne en valeur absolue ($MAPE$), à partir des formules suivantes :

$$RMSE = \sqrt{\frac{\sum_{i=1}^M (\hat{\theta}_i - \theta)^2}{M}}$$

$$MAPE = \frac{\sum_{i=1}^M |PE_i|}{M}$$

$$PE_i = \frac{\hat{\theta}_i - \theta}{\theta}$$

Avec $\hat{\theta}$ les paramètres estimés et θ les paramètres initiaux.

Nous avons également utilisé ce plan d'expérience pour comparer les résultats obtenus en utilisant une étape MCMC ou une étape SMC pour estimer ξ_x .

Puisque l'estimation de chacun de nos paramètres doit se rapprocher de notre valeur initiale, nous devrions avoir des estimations $\hat{\theta}$ autour de nos θ , soit des PE relativement proches de 0. À

partir des résultats obtenus pour les 1000 itérations, nous avons représenté sur la Figure 3.15 les boxplots des PE pour chacun des 7 paramètres et pour les deux méthodes de simulation de ξ_x .

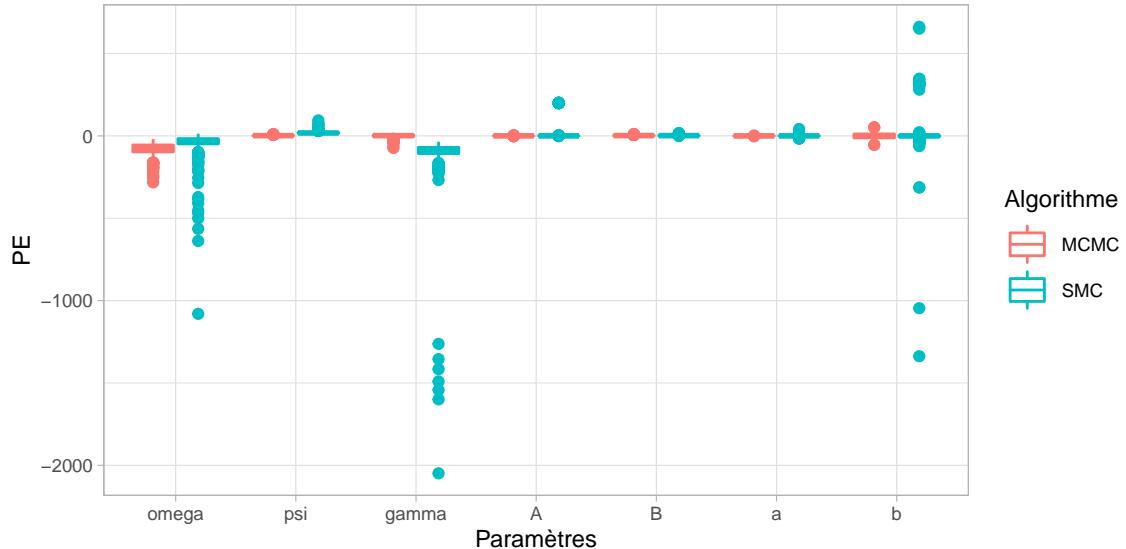


Figure 3.15: Boxplot des PE des 1000 estimations de chacun des paramètres.

Nous pouvons remarquer qu'avec la version SAEM-MCMC les PE de l'ensemble des paramètres, hormis ω , varient autour de 0. En effet, ω semble être sous-estimé et avoir une variation plus grande. Nous pouvons penser que cette variation est due au fait que la valeur de ω initiale de 0.01 est faible, ce qui rend sa source de bruit difficile à distinguer des autres, plus importantes, qui sont elles bien estimées.

En revanche, il semblerait qu'avec la méthode SAEM-SMC, l'estimation de ω est meilleure, mais celles de ψ et γ sont nettement moins bonnes. L'erreur réalisée sur l'estimation des paramètres A, B, a, b paraît similaire pour les deux approches d'estimation de ξ_x . Cependant, nous remarquons que les estimations des paramètres via la méthode SAEM-SMC varient plus et sont parfois très éloignées de la valeur réelle. En effet, ω et γ sont parfois sous-estimés de manière excessive sur certaines simulations, tandis que b est parfois fortement sur-estimé ou sous-estimé.

Table 3.1: Valeurs attendues, estimées en moyenne et erreurs associées de chacun des paramètres, en utilisant une étape MCMC dans l'algorithme SAEM.

	ω	ψ	γ	A	B	a	b
θ	0.010	0.951	0.098	0.500	-0.250	0.100	1.000
$\hat{\theta}$	0.018	0.932	0.097	0.499	-0.245	0.100	0.999
MAPE	79.859	2.143	6.153	0.486	2.100	0.467	13.911
RMSE	0.009	0.026	0.008	0.003	0.006	0.001	0.173

Au regard de la Table 3.1, il convient de relativiser l'erreur réalisée sur le paramètre ω avec la méthode SAEM-MCMC : la valeur attendue est 0.01 et celle estimée en moyenne est égale à 0.014. La *MAPE* peut sembler élevée, mais encore une fois, il faut observer qu'en valeur absolue les estimations sont proches de la valeur à estimer, comme l'indique la *RMSE*. Nous pouvons noter que c'est bien le paramètre b dont l'estimation est la plus éloignée de la valeur attendue, ce qui rejoint les observations faites à partir des Figures ?? et ??.

Table 3.2: Valeurs attendues, estimées en moyenne et erreurs associées de chacun des paramètres, en utilisant une étape SMC dans l'algorithme SAEM.

	ω	ψ	γ	A	B	a	b
θ	0.010	0.951	0.098	0.500	-0.250	0.100	1.000
$\hat{\theta}$	0.014	0.773	0.197	0.478	-0.246	0.100	0.980
MAPE	40.939	18.723	101.959	4.579	1.752	0.453	16.310
RMSE	0.007	0.188	0.158	0.145	0.006	0.002	0.766

La Table 3.2 confirme les observations visuelles : la *MAPE* des paramètres ψ et γ augmentent fortement en estimant ξ_x via un SMC par rapport à la méthode SAEM-MCMC, tandis que celle du paramètre ω diminue légèrement et que pour les autres paramètres nous n'observons pas de réelle différence.

Au regard de ces résultats, la version SAEM-MCMC semble être à privilégier par rapport à la SAEM-SMC. Cependant, nous avons noté que l'implémentation en **R** de l'algorithme avec l'étape SMC est bien plus rapide que celle avec l'étape MCMC : les 1000 répétitions du plan d'expérience SAEM-SMC ont été exécutées en un peu moins de 15 minutes, contre plus d'une heure pour le SAEM-MCMC. Nous avons identifié plusieurs raisons à cela :

- la convergence plus rapide de l'algorithme SAEM-SMC permettant de réduire le nombre d'itérations de 500 à 100,
- la possibilité de vectoriser l'ensemble des opérations sur les P particules de SMC, alors que les répétitions du MCMC doivent être réalisées séquentiellement.

Il pourrait donc être intéressant de tenter d'améliorer l'implémentation du SMC, et notamment en proposant une autre simulation des particules au temps i , ne dépendant pas uniquement des particules au temps $i - 1$ mais également des observations $Y_{1:i}$.

3.4 Conclusion

Les narvals possèdent une grande dent avec des capacités sensorielles très développées. Cette dent comporte des sillons, apparaissant au cours de leur croissance et créant des motifs sinusoïdaux variants avec les saisons. Elle pourrait ainsi être utile pour approximer la durée de vie des baleines. Notre objectif était donc d'estimer les paramètres θ d'une sinusoïde Y perturbée par un bruit gaussien et un processus d'Ornstein-Uhlenbeck ξ_x au moyen d'un algorithme SAEM. Nous avons

proposé deux implémentations de l'algorithme SAEM, l'une estimant le processus ξ_x par une étape MCMC et l'autre par une étape SMC.

Nous avons comparé les reconstructions $\hat{\xi}_x$ et \hat{Y} et les estimations des paramètres $\hat{\theta}$ données par les deux versions. Visuellement, les reconstructions \hat{Y} sont très semblables et très satisfaisantes. Celle de ξ_x proposée par l'algorithme SAEM-SMC est apparue légèrement meilleure que celle de SAEM-MCMC, les deux demeurant tout à fait fidèles à la réalisation originale du processus. L'estimation $\hat{\theta}$ des paramètres a donné des résultats assez différents entre les deux approches. Certains des paramètres sont mieux approchés par la version SAEM-MCMC, d'autres par la version SAEM-SMC.

Pour juger plus rigoureusement les estimations des paramètres du modèle sinusoïdal, nous avons répété 1000 fois chacune des deux procédures SAEM et nous nous sommes intéressés aux distributions des erreurs réalisées, ainsi qu'à leurs moyennes. L'algorithme SAEM-MCMC s'est alors révélé plus stable que le SAEM-SMC car souffrant d'une moins grande variance dans l'estimation des paramètres.

Cependant, le SAEM-SMC converge nettement plus rapidement que la version MCMC et son implémentation étant vectorisable, il demande un temps de calcul environ 5 fois moins important. Cela suggère qu'il pourrait être intéressant de tenter d'améliorer l'échantillonnage des particules générées par l'étape SMC pour perfectionner l'estimation des paramètres du modèle, tout en bénéficiant du temps de calcul restreint de l'algorithme SAEM-SMC.

Enfin, nous avons manipulé uniquement des données simulées, il reste donc à confronter les deux versions de l'algorithme à des données réelles, probablement plus bruitées !

Références

- [1] Outi M. Tervo et al. “Narwhals react to ship noise and airgun pulses embedded in background noise”. In: *Biology Letters* 17.11 (2021), p. 20210220. DOI: [10.1098/rsbl.2021.0220](https://doi.org/10.1098/rsbl.2021.0220). eprint: <https://royalsocietypublishing.org/doi/pdf/10.1098/rsbl.2021.0220>. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsbl.2021.0220>.
- [2] Wikipedia contributors. *Narval*. 2022. URL: <https://fr.wikipedia.org/wiki/Narval> (visited on 02/02/2023).
- [3] Jean-Pierre Sylvestre. *Dans l'Arctique canadien avec la licorne de mer*. 2018. URL: https://www.people-animal.com/article,lecture,1160_dans-l-arctique-canadien-avec-la-licorne-de-mer.html (visited on 02/02/2023).
- [4] Peter Bondo. *The narwhal's tusk reveals its past living conditions*. 2021. URL: <https://phys.org/news/2021-03-narwhal-tusk-reveals-conditions.html> (visited on 02/02/2023).
- [5] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum Likelihood from Incomplete Data Via the EM Algorithm”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22. DOI: <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>. eprint: <https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1977.tb01600.x>. URL: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x>.
- [6] Bernard Delyon, Marc Lavielle, and Eric Moulines. “Convergence of a Stochastic Approximation Version of the EM Algorithm”. In: *The Annals of Statistics* 27.1 (1999), pp. 94–128. URL: <http://www.jstor.org/stable/120120> (visited on 02/21/2023).
- [7] Estelle Kuhn and Marc Lavielle. “Coupling a stochastic approximation version of EM with an MCMC procedure”. In: *ESAIM: Probability and Statistics* 8 (2004), pp. 115–131. DOI: [10.1051/ps:2004007](https://doi.org/10.1051/ps:2004007). URL: <http://www.numdam.org/articles/10.1051/ps:2004007/>.
- [8] Sophie Donnet and Adeline Samson. “Using PMCMC in EM algorithm for stochastic mixed models: theoretical and practical issues”. In: *Journal de la société française de statistique* 155.1 (2014), pp. 49–72. URL: http://www.numdam.org/item/JJSFS_2014__155_1_49_0/.