CRISTIAN MEZA

Extensions et applications de l'algorithme SAEM pour les modèles mixtes

Thèses d'Orsay, 2006

<http://www.numdam.org/item?id=BJHTUP11_2006__0714__A1_0>

N° d'ordre: 2784

UNIVERSITE PARIS-SUD
FACULTE DES SCIENCES D'ORSAY

THESE

Présentée pour obtenir

LE GRADE DE DOCTEUR EN SCIENCES
DE L'UNIVERSITE PARIS XI

Spécialité : Mathématiques

par

Cristian MEZA

EXTENSIONS ET APPLICATIONS DE L'ALGORITHME SAEM POUR LES MODELES MIXTES

Soutenue le 8 décembre 2006 devant la commission d'examen :

M.	Alain BACCINI	(Rapporteur)
M.	Jean-Jacques DAUDIN	(Rapporteur)
M.	Jean-Louis FOULLEY	(Co-Directeur de thèse)
Mme	Elisabeth GASSIAT	(Présidente)
M.	Marc LAVIELLE	(Co-Directeur de thèse)

A mis padres

Remerciements

Je voudrais tout d'abord remercier mes co–directeurs de thèse Jean–Louis Foulley et Marc Lavielle pour la direction de cette thèse et leur soutien au long de ces années. Merci Marc pour avoir pris le temps de travailler avec moi, même si cela n'était pas prévu au début. Je te remercie aussi de m'avoir proposé de collaborer avec Jean–Louis Foulley. Un grand merci à Jean–Louis Foulley pour les nombreuses et brillantes idées que vous avez eu durant cette période de travail. Je vous suis très reconnaissant de m'avoir si bien accueilli à l'INRA et d'avoir su créer cette dynamique de groupe avec Florence Jaffrézic. J'en profite pour remercier Florence qui a joué un role très important dans cette collaboration avec l'INRA, merci.

Je tiens à remercier Alain Baccini et Jean–Jacques Daudin pour avoir accepté d'être rapporteur de cette thèse. Je les remercie pour leurs remarques pertinentes. Je remercie aussi Elisabeth Gassiat pour sa présence dans le jury.

Merci à tout ceux qui m'ont aidé d'une manière ou une autre durant mon séjour à Orsay. En particulier merci Mélanie, les discussions que nous avons eues étaient vraiment très sympas.

Je remercie mes "potes" de toujours, j'étais très content de vous revoir après ces quelques années. Merci à Boujemil, Farid, Mehdi, Mustapha et Youssef, mes frères, pour essayer de comprendre ce que je faisais ici.

También me gustaría darle las gracias a dos personas que me ayudaron a venir hacer mi tesis en Francia: primero muchas gracias profesor Rolando Rebolledo por su orientación y sus recomendaciones. Me ayudó a tomar decisiones, gracias. Un abrazo gigante a Pilar Iglesias por todo lo que hizo antes y durante mi tesis. Siempre le agradeceré, no habría llegado hasta acá sin usted.

Gracias a los amigos que me hice durante esta estadía. Más allá de hablar un mismo idioma creo que aprendimos a conocernos en estos años. Gracias Ana, Lisandro, Hector, Mario y Alison por su ayuda y apoyo. También le doy las gracias, por su gran ayuda, a mi "familia chilena" que lleva varios años viviendo por acá (gracias "tios"!!) al igual que a sus hijos y amigos de tantos años, Carlos y Osvaldo.

Obviamente le doy las gracias a mis queridos "viejos", que me apoyaron a la distancia. Creo que valió la pena todo esto. Los quiero. No me olvido de mis "hermanitas" que hasta me vinieron a ver. Gracias por todo. Bueno para que no se pongan celosos, también tendré que nombrar a mis lindos sobrinos, Eva y Thomas. Gracias por la visita, ojalá tengamos más tiempo ahora para pasarlo juntos, tenemos que recuperar el tiempo perdido.

Y finalmente, le doy las gracias a Natalia por haber estado allí siempre, en las buenas y sobre todo en las malas. Te quiero.

Cette thèse a été financée par l'ambassade de France au Chili et le gouvernement chilien à travers CONICYT.

Table des matières

Liste des tableaux

Table des figures

Chapitre 1

Introduction

Contents

1.1 Sommaire

Dans cette thèse, nous nous intéressons à l'estimation paramètrique dans le cadre des modèles mixtes. Durant les trois premiers chapitres, on s'intéresse aux modèles mixtes non linéaires et dans le dernier chapitre nous étudions le cas des modèles linéaires généralisés à effets mixtes (GLMM[1]), nous limitant au modèle *probit* mixte pour des données binaires. Ces différents modèles sont traités comme des modèles à données manquantes et nous utilisons une version stochastique de l'algorithme itératif EM (*Expectation Maximization*) pour estimer les paramètres inconnus de ces modèles.

Dans le second chapitre de ce manuscrit, nous adaptons l'algorithme SAEM (*"Stochastic Approximation EM"*) pour l'appliquer dans le contexte de la génétique animale, et plus précisément les courbes de croissance. Le modèle étudié ici est un modèle mixte non linéaire qui en plus des effets aléatoires individuels, tient compte des effets génétiques liés aux animaux. Cela revient à rajouter un autre niveau de complexité au modèle en augmentant le nombre de paramètres à estimer. Deux analyses de courbe de croissance sont présentées, la première portant sur le poids chez le boeuf et la seconde portant sur le poids chez le poulet, toutes deux impliquant des données expérimentales collectées à l'Institut National de Recherche Agronomique (INRA).

Dans le troisième chapitre, nous proposons une extension de l'algorithme SAEM qui permet d'augmenter la vitesse de convergence de l'algorithme et d'éviter des maxima locaux de la vraisemblance dans certains cas. Pour cela nous avons adapté l'algorithme PX–EM. Ce nouvel algorithme, que nous nommons PX–SAEM (*"Parameter Expansion"* version), "élargit" le modèle en introduisant un paramètre auxiliaire ce qui a pour conséquence d'accélerer la convergence vers les estimateurs de Maximum de Vraisemblance. Nous proposons d'utiliser cette stratégie seulement durant les premières itérations de l'algorithme, puisque c'est là que l'on observe le gain de vitesse, puis de revenir à l'algorithme SAEM standard. Ceci permet de maintenir les propriétés de convergence de l'algorithme SAEM. Nous détaillons également comment appliquer cet algorithme au cas particulier des modèles mixtes linéaires et non linéaires. Deux applications permettent d'illustrer les performances de ce nouvel algorithme dans le cadre des modèles mixtes linéaires et non linéaires, montrant le gain de vitesse obtenu mais aussi le fait que le PX–SAEM permet d'éviter des maxima locaux.

Dans le quatrième chapitre, nous proposons une nouvelle procédure d'estimation des composantes de variance dans les modèles non linéaires mixtes, utilisant pour cela la méthode d'estimation du Maximum de Vraisemblance Restreinte (REML). Cette méthode

1. De l'anglais "Generalized Linear Mixed Models"

est mise en oeuvre grâce à l'algorithme SAEM. Trois applications sont présentées, la première est relative à un modèle linéaire mixte portant sur des données réelles d'ultrafiltration. Cette exemple nous sert à valider notre algorithme puisque nous pouvons, dans ce cas, comparer les résultats à ceux d'un EM analytique. La seconde application est un premier modèle mixte non linéaire portant sur des données réelles de croissance d'arbre. La dernière application étudiée porte sur une courbe de croissance non linéaire chez le poulet. Une étude de simulation a été réalisée sur ces données montrant que notre procédure REML permet de réduire de manière significative le biais des estimations de variance en comparaison avec les estimations obtenues par Maximum de Vraisemblance.

Le cinquième chapitre porte sur l'application de l'algorithme SAEM dans le cadre des Modèles Linéaires Généralisés Mixtes (GLMM). Nous étudions ici le modèle probit mixte pour analyser des données binaires. Nous détaillons comment appliquer, dans ce contexte, l'algorithme SAEM mais aussi les extensions que nous proposons dans les chapitres antérieurs comme l'algorithme PX–SAEM ainsi que la technique SAEM–REML. Plusieurs applications sont présentées pour illustrer ces méthodes.

1.2 Le modèle mixte gaussien

Les modèles mixtes sont des modèles qui décrivent des données, généralement répétées (i.e. on observe plus d'une observation par sujet ou individu), en considérant à la fois des paramètres associés à la population et des paramètres liés aux individus, permettant ainsi d'identifier les différentes sources de variation présentes dans les données. Cette approche qui exploite la variabilité entre et intra–populations trouve des applications dans de nombreux secteurs tels que l'agronomie, la génétique animale, la pharmacologie ou les neurosciences.

On considére le modèle suivant:

$$y_{ij} = g(x_{ij},\boldsymbol{\phi_i}) + h(x_{ij},\boldsymbol{\phi_i})\varepsilon_{ij}, \quad 1 \leq i \leq N, 1 \leq j \leq n_i, \tag{1.2.1}$$

où $y_{ij} \in \mathbb{R}$ représente l'observation j du sujet i, (x_{ij}) est un ensemble connu de variables de regression (le temps), N est le nombre total de sujets et n_i est le nombre d'observations du sujet i. Les résidus du modèle (ε_{ij}) sont supposés être indépendants et identiquement distribués (i.i.d.) gaussiens de moyenne nulle et de variance σ^2. Le vecteur individuel aléatoire $\boldsymbol{\phi_i} \in \mathbb{R}^d$ a la forme suivante:

$$\boldsymbol{\phi_i} = \boldsymbol{X_i\beta} + \boldsymbol{\eta_i} \qquad \text{avec} \quad \boldsymbol{\eta_i} \sim_{i.i.d.} \mathcal{N}(0,\boldsymbol{\Gamma}), \tag{1.2.2}$$

où β représente les paramètres inconnus de population (*effets fixes*), $\boldsymbol{X_i}$ est une matrice d'incidence de covariables connue et η_i les variables aléatoires individuelles supposées i.i.d. $\mathcal{N}(0,\Gamma)$. On suppose de plus que les (ε_{ij}) et les (η_i)(*effets aléatoires*) sont mutuellement indépendants.

On parle de modèle mixte gaussien lorsque l'on considère que les effets aléatoires (η_i) et les résidus (ε_{ij}) sont gaussiens.

On dira que le modèle mixte est linéaire si g est une fonction linéaire de ϕ_i et $h = 1$, et à l'inverse, on dira que le modèle mixte est non linéaire si g est une fonction non linéaire de ϕ_i et $h \neq 1$. On remarque que les (y_{ij}) sont gaussiens dans les cas d'un modèle linéaire mixte gaussien.

Un objectif est donc d'estimer les effets fixes mais aussi les composantes de variance présents dans le modèle, cela revient à chercher à estimer le vecteur de paramètres inconnus $\theta = (\beta,\Gamma,\sigma^2)$.

1.3 Le modèle linéaire généralisé mixte

Toutes les données ne sont pas modélisables par la loi normale (par exemple l'analyse de données discrètes et de survie). Quand les données sont modélisées par des lois qui appartiennent à la famille exponentielle (lois normale, Poisson, binomiale, Bernoulli, exponentielle, gamma...), les Modèles Linéaires Généralisés (GLM[2]) et leurs extensions les Modèles Linéaires Généralisés Mixtes (GLMM), peuvent être utilisés. Cette classe de modèles généralise les modèles linéaires classiques en termes de loi de probabilité et aussi en termes de lien à la linéarité.

D'après McCullagh and Nelder (1989), trois hypothèses caractérisent un GLM: un composant aléatoire (distribution du vecteur réponse), un composant systématique (la fonction linéaire des covariables) et une fonction de lien.

- **La distribution du vecteur réponse**

Une nouvelle fois, nous nous plaçons dans le contexte des données répétées. Les éléments du vecteur réponse $\boldsymbol{y_i} = \{y_{ij}\}$, avec $i = 1,\cdots,N$ et $j = 1,\cdots,n_i$ sont supposés indépendants avec une distribution qui appartient à la famille exponentielle (voir

2. De l'anglais "Generalized Linear Models"

Nelder and Lee, 1992):

$$f(\boldsymbol{y}_i;\theta_i,\psi_i) = \exp\left\{\frac{\boldsymbol{y}_i\theta_i - b(\theta_i)}{a(\psi_i)} + c(\boldsymbol{y}_i,\psi_i)\right\} \tag{1.3.3}$$

où $a(\cdot)$, $b(\cdot)$ est $c(\cdot)$ sont des fonctions spécifiques à chaque distribution et avec $a(\psi_i) = \psi/w_i$, où ψ est un paramètre de dispersion et w_i un poids connu. Chaque membre de la famille exponentielle est identifiable par sa moyenne $\mu_i = b'(\theta_i)$ et sa variance $\nu(\mu_i) = b''(\theta_i)a(\psi_i)$.

- **La fonction linéaire des covariables**

Le composant systématique est une fonction linéaire des covariables où le prédicteur linéaire a la forme suivante

$$\boldsymbol{\omega}_i = \boldsymbol{X}_i\boldsymbol{\beta},$$

où $\boldsymbol{X}_i$ est une matrice connue de dimension $n_i \times p$ et $\boldsymbol{\beta}$ est un vecteur inconnu de dimension $p \times 1$.

- **La fonction de lien**

La fonction de lien $l(\cdot)$ est une fonction strictement monotone différenciable qui relie la valeur espérée de la distribution du vecteur réponse μ_i au prédicteur linéaire $\boldsymbol{\omega}_i$:

$$\boldsymbol{\omega}_i = l(\boldsymbol{\mu}_i).$$

Si la réponse suit une distribution normale et $l(\boldsymbol{\mu}) = \boldsymbol{\mu}$, on retrouve le modèle de régression linéaire.

De même que les effets aléatoires ont été introduits dans les modèles linéaires, il est naturel de considérer des effets individuels dans le contexte du GLM obtenant ainsi le GLMM.

Il y a diverses façon de présenter la version mixte du GLM. L'une d'entre elles revient à faire appel à une approche hiérarchique:

1. conditionnellement aux effets aléatoires $\boldsymbol{\eta_i}$, les données $\boldsymbol{y_i}$ sont décrites par un GLM standard;

2. les effets aléatoires $\boldsymbol{\eta_i}$ sont distribués classiquement comme dans un modèle mixte linéaire gaussien (voir 1.2.2).

Plus en détails, on ajoute des effets aléatoires $\eta_i \in \mathbb{R}^d$ au prédicteur linéaire de la manière suivante:

$$\omega_i = X_i\beta + Z_i\eta_i \tag{1.3.4}$$

où Z_i est une matrice connue de dimension $n_i \times d$ et les effets aléatoires η_i sont i.i.d. $\mathcal{N}(0,\Gamma)$.

Conditionnellement à η_i, le GLMM a les mêmes propriétés que le GLM, c'est–à–dire:

1. la distribution conditionnelle de y_i sachant η_i est un GLM avec le prédicteur linéaire définit en (1.3.4);

2. les composantes du vecteur réponse y_i sont conditionnellement indépendantes sachant η_i;

3. l'espérance de la distribution conditionnelle de y sachant les effets aléatoires est associée au prédicteur linéaire par la fonction de lien:

$$\omega_i = l(\mu_i) \qquad \text{où} \quad \mu_i = \mathbb{E}(y_i|\eta).$$

Dans ce modèle, notre objectif est d'estimer le vecteur de paramètre $\theta = (\beta,\Gamma)$ et donc en déduire les μ_i.

1.4 L'algorithme EM et ses extensions

Les modèles mixtes étudiés ici peuvent être traités comme un problème à données manquantes. Les données observées $y \sim q(y;\theta)$, où $y \in \mathbb{R}^N$, sont en fait des observations partielles des données complètes $(y,\phi) \sim f(y,\phi;\theta)$, où $\phi \in \mathbb{R}^{N\times d}$, et f, q sont des distributions de densités connues.

Une méthode usuelle pour estimer θ est d'utiliser l'estimateur du Maximum de Vraisemblance. Cette procédure d'estimation pour θ consiste à calculer la valeur de $\hat{\theta}$ qui maximise la vraisemblance observée q, c'est–à–dire:

$$\begin{aligned} q(y;\theta) &= E[f(y,\phi;\theta)] = \int f(y,\phi;\theta)d\mu(\phi) \\ \hat{\theta} &= \arg\max_{\theta} q(y;\theta). \end{aligned} \tag{1.4.5}$$

Dans le modèle mixte, il est difficile de calculer l'estimateur du Maximum de Vraisemblance directement à partir de l'équation (1.4.5) du fait de l'intrégation par rapport à $\boldsymbol{\phi}$. Dempster et al. (1977) proposent de remplacer dans cette équation $q(\boldsymbol{y};\boldsymbol{\theta})$ par son espérance conditionnelle $E[q(\boldsymbol{y};\boldsymbol{\theta})|\boldsymbol{y}]$ par rapport aux données observées $\boldsymbol{y}$. Comme l'intrégration par rapport à la densité de $\boldsymbol{\phi}|\boldsymbol{y}$ suppose les paramètres $\boldsymbol{\theta}$ connus, Dempster et al. (1977) proposent de procèder par itération en intégrant par rapport à $\boldsymbol{\phi}|\boldsymbol{y};\boldsymbol{\theta}=\boldsymbol{\theta_k}$ où $\boldsymbol{\theta_k}$ est la valeur courante du paramètre $\boldsymbol{\theta}$ à l'itération k. C'est l'idée de base de l'algorithme EM (*"Expectation Maximization"*).

L'algorithme EM est un algorithme itératif qui permet de calculer les estimateurs du Maximum de Vraisemblance générant, à partir d'un point initial $\boldsymbol{\theta}_0$, une suite $\{\boldsymbol{\theta}_k\}$ d'estimateurs, $k = 1,2,3,\cdots$. Chaque itération k est composée des deux étapes suivantes: dans une première étape on calcule l'espérance conditionnelle de la log–vraisemblance complète par rapport aux données complètes et à la valeur courante du paramètre $\boldsymbol{\theta}$ (étape E); ensuite, durant la seconde étape, on réactualise $\boldsymbol{\theta}$ en maximisant cette quantité en $\boldsymbol{\theta}$ (étape M).

Donc, si on se situe à la itération k, l'algorithme EM, d'une manière générale est structuré de la manière suivante:

- *Etape E*: on évalue $Q_k(\boldsymbol{\theta})$ qui est défini comme il suit

$$Q_{k+1}(\boldsymbol{\theta}) = E(\log f(\boldsymbol{y},\boldsymbol{\phi};\boldsymbol{\theta}_k)|\boldsymbol{y};\boldsymbol{\theta}_k).$$

- *Etape M*: on réactualise la valeur courante de $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} Q_{k+1}(\boldsymbol{\theta}).$$

L'algorithme EM peut se simplifier si on suppose que la vraisemblance complète $f(\boldsymbol{y},\boldsymbol{\phi};\boldsymbol{\theta})$ appartient à la famille exponentielle, c'est à dire que l'on a

$$f(\boldsymbol{y},\boldsymbol{\phi};\boldsymbol{\theta}) = \exp\left\{-\Psi(\boldsymbol{\theta}) + \langle\tilde{S}(\boldsymbol{y},\boldsymbol{\phi}),\Phi(\boldsymbol{\theta})\rangle\right\} \tag{1.4.6}$$

où $\langle\cdot,\cdot\rangle$ représente le produit scalaire et où $\tilde{S}(\boldsymbol{y},\boldsymbol{\phi})$ sont les statistiques minimales du modèle complet.

Sous cette hypothèse, l'algorithme EM, à l'itération k, se ramène à la forme suivante

- *Etape E*: on évalue la quantité $s_{k+1} = E[\tilde{S}(\boldsymbol{y},\boldsymbol{\phi})|\boldsymbol{y};\boldsymbol{\theta}_k]$.
- *Etape M*: on réactualise la valeur courante de $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} \{-\Psi(\boldsymbol{\theta}) + \langle s_{k+1}, \Phi(\boldsymbol{\theta})\rangle\}.$$

Wu (1983) prouve la convergence de la suite $(\boldsymbol{\theta}_k)$ vers un point de stationnarité de la log–vraisemblance des observations, sous des hypothèses générales de régularité du modèle. Delyon et al. (1999) obtiennent un résultat de convergence de l'algorithme EM dans le cadre des modèles appartenant à la famille exponentielle. Ils supposent les hypothèses suivantes:

(EM1) Le modèle satisfait l'équation (1.4.6) où $\tilde{S}$ est une fonction borélienne sur $\mathbb{R}^l$ à valeurs dans $\mathcal{S}$ un ouvert de $\mathbb{R}^m$ telle que $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$

$$\int_{\mathbb{R}^l} |\tilde{S}(\boldsymbol{y},\boldsymbol{\phi})| p(\boldsymbol{\phi}|\boldsymbol{y};\boldsymbol{\theta}) d\boldsymbol{\phi} < \infty.$$

(EM2) Les fonctions ψ et Φ sont deux fois continûment différentiables sur $\boldsymbol{\Theta}$.

(EM3) On définit la fonction $\bar{s} : \boldsymbol{\Theta} \longrightarrow \mathcal{S}$ de la manière suivante:

$$\bar{s}(\boldsymbol{\theta}) = \int_{\mathbb{R}^l} \tilde{S}(\boldsymbol{y},\boldsymbol{\phi}) p(\boldsymbol{\phi}|\boldsymbol{y};\boldsymbol{\theta}) d\boldsymbol{\phi}$$

et on la considère différentiable sur $\boldsymbol{\Theta}$.

(EM4) On suppose que la log–vraisemblance des données observées l est continûment différentiable sur $\boldsymbol{\Theta}$ et aussi que

$$\partial_\theta \int_{\mathbb{R}^l} f(\boldsymbol{y},\boldsymbol{\phi};\theta) d\boldsymbol{\phi} = \int_{\mathbb{R}^l} \partial_\theta f(\boldsymbol{y},\boldsymbol{\phi};\theta) d\boldsymbol{\phi}.$$

(EM5) On définit la fonction $L : \mathcal{S} \times \boldsymbol{\theta} \to \mathbb{R}$ de la manière suivante:

$$L(s;\boldsymbol{\theta}) = -\psi(\boldsymbol{\theta}) + \langle s, \Phi(\boldsymbol{\theta})\rangle$$

alors il existe une fonction $\hat{\theta} : \mathcal{S} \to \boldsymbol{\Theta}$ telle que

$$\forall s \in \mathcal{S}, \quad \forall \boldsymbol{\theta} \in \boldsymbol{\Theta}, \quad L(s;\hat{\boldsymbol{\theta}}(s)) \geq L(s;\boldsymbol{\theta}).$$

Théorème 1.4.1 *On suppose les hypothèses (EM1)–(EM5) vraies, donc pour tout point initial $\theta_0 \in \Theta$, la suite $(l(\theta_k))$ obtenue avec l'algorithme EM est une suite croissante et l'algorithme converge vers un point stationnaire de la vraisemblance.*

En pratique, pour se détacher du caractère local du maximum atteint, on fait tourner l'algorithme EM un grand nombre de fois à partir de valeurs initiales différentes de manière à avoir de plus grandes chances d'atteindre le maximum global de vraisemblance.

L'algorithme EM présente plusieurs inconvénients, notamment la complexité de l'étape du calcul d'espérance et de l'étape de maximisation, mais aussi la lenteur de convergence dans certains cas. Ces limitations ont donnés lieu à plusieurs extensions et variations (voir McLachlan and Krishnan, 1997).

Par exemple, Meng and Rubin (1993a) remplacent l'étape M par une suite de maximisations sous contraintes de la log–vraisemblance complète. Ce nouvel algorithme est appelé "Expectation Conditional Maximisation" (ECM). Fesslern and Hero (1994) proposent l'algorithme SAGE (space-alternating generalized EM), qui a pour but de réduire le temps de convergence de l'algorithme EM. L'idée est de diviser l'espace des données complètes en plusieurs sous espaces cachés moins informatifs et de les estimer de façon séquentielle. Liu et al. (1998) proposent une technique pour augmenter la vitesse de convergence de l'algorithme EM, l'algorithme PX–EM. Il s'agit d'"élargir" le modèle incluant un paramètre de "travail" ce qui a pour conséquence d'accélérer les algorithmes de types EM.

Les versions stochastiques de l'algorithme EM ont été introduites pour traiter des situations où l'étape E n'est pas réalisable de manière exacte. Celeux and Diebolt (1985) proposent une première version stochastique de l'algorithme EM, algorithme nommée SEM (Simulated EM), où une étape de simulation est rajoutée (étape S). A l'itération k, on simule une réalisation des données manquantes $\boldsymbol{\phi}^{(k+1)}$ à partir de la loi conditionnelle $p(\cdot|\boldsymbol{y};\boldsymbol{\theta}_k)$. On maximise ensuite en $\boldsymbol{\theta}$ la log–vraisemblance complète $\log f(\boldsymbol{y},\boldsymbol{\phi}^{(k+1)};\boldsymbol{\theta})$.

L'algorithme Monte Carlo EM (MCEM) remplace l'étape S de l'algorithme SEM par une étape d'approximation de Monte Carlo qui se base sur un grand nombre de simulations indépendantes des données manquantes, voir Wei and Tanner (1990a).

Une extension de MCEM est proposée par Jank (2004), en se basant sur des méthodes de Quasi–Monte Carlo. Ces méthodes génèrent des suites déterministes de points qui peuvent améliorer de manière significative l'efficacité des approximations Monte Carlo basées sur des échantillons complètement aléatoires. Cet algorithme est nommé QMCEM.

Un inconvenient majeur de ces méthodes est qu'elles consomment beaucoup de temps de calcul dû au grand nombre de simulations qu'elles impliquent. Delyon et al. (1999)

proposent une nouvelle version stochastique, l'algorithme SAEM (Stochastic Approximation EM), qui permet de réduire le nombre de simulations nécessaires en conservant de bonnes propriétés de convergence.

1.5 L'algorithme SAEM et ses extensions

A chaque itération, l'algorithme SAEM remplace l'étape E de l'algorithme EM par deux étapes, tout d'abord une étape de Simulation des données manquantes, étape S, et ensuite une étape d'approximation stochastique, étape A.

Donc si on se situe à l'itération k, durant l'étape S on simule une réalisation des données manquantes $\boldsymbol{\phi}^{(k+1)}$ à partir de la loi conditionnelle $p(\cdot|\boldsymbol{y};\boldsymbol{\theta}_k)$ et on réalise ensuite l'approximation stochastique de la manière suivante

$$Q_{k+1}(\boldsymbol{\theta}) = Q_k(\boldsymbol{\theta}) + \gamma_k \left(\log\, f(\boldsymbol{y},\boldsymbol{\phi}^{(k+1)};\boldsymbol{\theta}) - Q_k(\boldsymbol{\theta}) \right), \tag{1.5.7}$$

où (γ_k) est une suite décroissante de pas positifs. L'étape de maximisation est elle inchangée.

Si on suppose que le modèle appartient à la famille exponentielle, c'est à dire que l'on a (1.4.6), l'algorithme SAEM est beaucoup plus simple puisqu'on réalise l'approximation stochastique (1.5.7) sur une statistique exhaustive $\tilde{S}$ du modèle. L'algorithme SAEM peut donc s'écrire de la manière suivante:

- *Etape S*: on simule $\boldsymbol{\phi}^{(k+1)}$ à partir de la loi conditionnelle $p(\cdot|\boldsymbol{y};\boldsymbol{\theta}_k)$.

- *Etape A*: on a construit une suite (s_k), initialisée par s_0, de la manière suivante:

$$s_{k+1} = s_k + \gamma_k \left[\tilde{S}(\boldsymbol{y},\boldsymbol{\phi}^{(k+1)}) - s_k \right]. \tag{1.5.8}$$

- *Etape M*: on réactualise la valeur courante de $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} \left\{ -\Psi(\boldsymbol{\theta}) + \langle s_{k+1},\boldsymbol{\phi}^{k+1}),\Phi(\boldsymbol{\theta}) \rangle \right\}. \tag{1.5.9}$$

Delyon et al. (1999) ont obtenu un résultat de convergence pour l'algorithme SAEM en considérant les hypothèses **(EM1)**–**(EM5)** et des hypothèses sur les données manquantes

simulées et la suite de pas (γ_k). On suppose que les variables aléatoires $s_0,\phi^{(1)},\phi^{(2)},\cdots$ sont définis sur le même espace de probabilité $(\Omega,\mathcal{A},P)$. Soit $\mathcal{F} = \{\mathcal{F}_k\}_{k\geq 0}$ la famille croissante de tribus engendrés par les variables aléatoires $\phi^{(1)},\phi^{(2)},\cdots,\phi^{(k)}$. On a de plus les hypothèses suivantes:

(SAEM1) Pour tout k dans $\mathbb{N}$, $\gamma_k \in [0,1]$, $\sum_{k=1}^{\infty}\gamma_k = \infty$ et $\sum_{k=1}^{\infty}\gamma_k^2 < \infty$.

(SAEM2) $l : \boldsymbol{\theta} \to \mathbb{R}$ et $\hat{\boldsymbol{\theta}} : \mathcal{S} \to \Theta$ est m fois différentiables.

(SAEM3) 1. Pout toute fonction Φ borélienne positive on a:

$$E[\Phi(\boldsymbol{\phi}^{(k+1)})|\mathcal{F}_k] = \int \Phi(\boldsymbol{\phi})p(\boldsymbol{\phi}|\boldsymbol{y};\theta_k)d\boldsymbol{\phi}.$$

2. Pour tout $\boldsymbol{\theta} \in \Theta$, $\int ||\tilde{S}(\boldsymbol{y},\boldsymbol{\phi})||^2 p(\boldsymbol{\phi}|\boldsymbol{y};\theta_k)d\boldsymbol{\phi} < \infty$, et la fonction

$$\Gamma(\boldsymbol{\theta}) = \mathrm{Cov}_{\theta}(\tilde{S}(\boldsymbol{y},\boldsymbol{\theta}))$$

est continue en $\boldsymbol{\theta}$.

Théorème 1.5.1 *On suppose les hypothèses (EM1)-(EM5) et (SAEM1)-(SAEM3) vraies, et que la suite $(s_k)_{k\geq 0}$ est à valeurs dans un sous-ensemble compact de $\mathcal{S}$, donc la suite $(\boldsymbol{\theta_k})_{k\geq 0}$ obtenue avec SAEM converge vers une valeur stationnaire de la log-vraisemblance observée f.*

Une restriction importante de l'algorithme original SAEM est que l'on doit connaître la loi conditionnelle $p(\cdot|\boldsymbol{y};\boldsymbol{\theta})$ pour réaliser l'étape S, ce qui en pratique n'est généralement pas le cas. C'est pour cela que Kuhn and Lavielle (2004) proposent de combiner l'algorithme SAEM avec les méthodes Monte Carlo par chaînes de Markov (MCMC) pour réaliser l'étape de simulation sans connaître de manière explicite cette loi conditionnelle.

Lorsque la simulation à partir de la distribution a posteriori $p(\cdot|\boldsymbol{y};\boldsymbol{\theta})$ n'est pas réalisable, Kuhn and Lavielle (2004) proposent de considérer une chaîne de Markov ergodique de probabilité de transition $\Pi_{\boldsymbol{\theta}}$ ayant pour loi stationnaire cette loi de distribution a posteriori. Ils proposent de faire évoluer en même temps la convergence de la chaîne issue de la méthode de Monte Carlo vers la loi stationnaire $p(\cdot|\boldsymbol{y};\boldsymbol{\theta})$ et celle de la suite (θ_k).

Donc, à l'itération k et dans le contexte des modèles de type exponentiel, l'algorithme SAEM–MCMC est constitué des trois étapes suivantes:

- *Etape S*: en utilisant $\phi^{(k)}$, on simule une réalisation $\phi^{(k+1)}$ à partir de la probabilité de transition $\Pi_{\theta_k}(\phi^{(k)},\cdot)$.

- *Etape A*: on réactualise s_k en utilisant (1.5.8).

- *Etape M*: on réactualise la valeur courante de θ en utilisant (1.5.9).

D'un point de vue pratique, Kuhn and Lavielle (2004) proposent d'utiliser l'algorithme Hasting–Metropolis pour générer cette chaîne de Markov. Cette méthode considère une probabilité de transition q_c dite "loi instrumentale" et génère, à l'itération k, un "candidat" ϕ^c selon $q_c(\phi^{(k)},\cdot)$, où $\phi^{(k)}$ est la valeur de la chaîne à l'itération k. On accepte le candidat, c'est à dire $\phi^{(k+1)} = \phi^c$, avec la probabilité $p_{HM} = \min\left(\frac{p(\phi^c|y)}{p(\phi^{(k)}|y)}\frac{q_c(\phi^c,\phi^{(k)})}{q_c(\phi^{(k)},\phi^c)},1\right)$ sinon $\phi^{(k+1)} = \phi^{(k)}$ avec la probabilité $1 - p_{HM}$.
Les auteurs obtiennent la convergence presque sûr de la suite (θ_k) vers un maximum local de la log–vraisemblance observée, en considérant certaines hypothèses sur la chaîne de Markov de transition Π_θ.

1.6 L'analyse génétique de courbe de croissance en utilisant l'algorithme SAEM

Dans le second chapitre de cette thèse, nous nous intéressons à une application de l'algorithme SAEM dans le domaine de l'analyse génétique de courbe de croissance animale. Les courbes de croissance permettent de décrire un processus complet de croissance d'un animal en fonction de quelques paramètres qui ont une interprétation biologique. Elles reflètent la corrélation entre l'impulsion inhérente d'un animal pour se développer et grandir, et l'environnement dans lesquel ces impulsions sont exprimées. L'environnement est composé par exemple par le niveau individuel de production, la quantité et la qualité d'aliment consommées ou encore l'effort requis pour consommer et digérer cet aliment.

Généralement ces courbes de croissance sont modélisées par des fonctions non linéaires qui dépendent de l'âge de l'animal, où chaque paramètre a une interprétation biologique.

Les courbes non linéaires les plus usuelles pour expliquer la relation taille–âge sont des cas particuliers de la fonction dite de Richards (voir Fitzhugh, 1976):

$$y_t = \begin{cases} A[1 + B\exp(-Kt)]^m & \text{si } m < 0 \\ A[1 - B\exp(-Kt)]^m & \text{si } m > 0 \end{cases}, \tag{1.6.10}$$

où y_t est la taille ou le poids de l'animal à l'âge ou au temps t. On peut citer par exemple les fonctions monomoléculaire (Brody, 1945), Logistic, Gompertz (Laird, 1966) et Bertalanffy (Bertalanffy, 1957) – voir Tableau 1.1.

TAB. 1.1 – *Equations des courbes de croissance et relation avec la fonction de Richards*

Modèle	Equation y_t à l'âge t	m
Richards	$A[1 \pm B\exp(-Kt)]^m$	Variable
Gompertz	$A\exp(-Bexp(-Kt))$	$m \longrightarrow \infty$
Logistic	$A[1 + B\exp(-Kt)]^{-1}$	-1
Brody	$A[1 - B\exp(-Kt)]$	1
Bertalanfly	$A[1 - B\exp(-Kt)]^3$	3

Nous considérons dans cette étude le modèle hiérarchique suivant: pour l'animal i, on observe le poids y_{ij} à l'âge t_{ij}, avec $i = 1, \cdots, N$ et $j = 1, \cdots, n_i$. Un premier niveau décrit la courbe de croissance, utilisant pour cela une des fonctions décrites dans le Tableau 1.1. Un second niveau décrit la variabilité individuelle, considérant un modèle linéaire qui inclut les effets environnementaux et les effets génétiques.

D'une manière générale, le modèle s'écrit de la forme suivante:

$$y_{ij} = g(t_{ij}, \boldsymbol{\phi_i}) + h(t_{ij}, \boldsymbol{\phi_i})\varepsilon_{ij} \tag{1.6.11}$$

où $y_{ij} \in \mathbb{R}$ représente l'observation j du sujet i, à l'âge t_{ij}. Les résidus (ε_{ij}) sont supposées *i.i.d.* gaussiens avec une moyenne nulle et un variance inconnue σ^2.

Ici, g est une fonction non linéaire définie dans le Tableau 1.1. Le vecteur colonne $\boldsymbol{\phi_i}$ de $\mathbb{R}^d$, pour chaque individu i, se décompose de la manière suivante:

$$\boldsymbol{\phi_i} = \boldsymbol{X_i\beta} + \boldsymbol{Z_i u} + \boldsymbol{\eta_i} \tag{1.6.12}$$

où $\boldsymbol{\beta}$ est le vecteur des effets fixes du modèle, $\boldsymbol{u}$ représente les effets génétiques et $\boldsymbol{\eta_i}$ les effets environnementaux permanents. $\boldsymbol{X_i}$ et $\boldsymbol{Z_i}$ sont des matrices d'incidence connues.

On suppose que $\boldsymbol{u}$, de dimension $N_a d \times 1$, suit une distribution gaussienne de moyenne nulle et de variance $\boldsymbol{A} \otimes \boldsymbol{G}$, où la matrice $\boldsymbol{G}$, de dimension $d \times d$, représente la matrice de variance–covariance génétique des effets individuels, et la matrice $\boldsymbol{A}$ est une matrice connue qui correspond ici à la matrice de parenté entre les individus. Le vecteur $\boldsymbol{\eta_i}$ est aussi supposé normal, avec une moyenne nulle et une matrice de variance–covariance $\boldsymbol{P}$, de dimension $d \times d$.
L'objectif est donc d'obtenir les estimateurs par Maximum de Vraisemblance de $\boldsymbol{\theta} = (\boldsymbol{\beta},\boldsymbol{G},\boldsymbol{P},\sigma^2)$.

Dans la littérature ces modèles ont été étudiés auparavant, proposant par exemple l'utilisation de l'échantilleur de Gibbs dans un cadre Bayésien (Blasco et al., 2003). Mais ces méthodes présentent plusieurs inconvénients comme par exemple le choix des distributions a priori et des temps de calcul très grands.

D'un point de vue classique, McCulloch (1997) propose d'utiliser un algorithme hybride qui combine l'algorithme MCEM (Wei and Tanner, 1990a) et une méthode de Monte Carlo pour intégrer et maximiser la vraisemblance (MCMLE, voir Geyer, 1994). Là encore, ces techniques utilisent beaucoup de temps de calcul.
Nous proposons donc une extension de l'algorithme SAEM–MCMC proposée par Kuhn and Lavielle (2004) pour obtenir les estimateurs par Maximum de Vraisemblance des paramètres $\boldsymbol{\theta}$ applicable à ce type de modèle non linéaire de croissance animale. L'idée centrale est de considérer les données manquantes comme le vecteur $\boldsymbol{z} = (\boldsymbol{\phi},\boldsymbol{u})$ et appliquer l'algorithme SAEM. Pour réaliser l'étape de simulation, nous considérons un schéma de Gibbs, c'est à dire que si l'on se situe à l'itération k, nous allons simuler $\boldsymbol{\phi}^{(\boldsymbol{k+1})}$ à partir de la distribution conditionnelle $p(\cdot|\boldsymbol{y},\boldsymbol{u}^{(\boldsymbol{k})};\theta_k)$ et ensuite $\boldsymbol{u}^{(\boldsymbol{k+1})}$ à partir de la distribution conditionnelle $p(\cdot|\boldsymbol{y},\boldsymbol{\phi}^{(\boldsymbol{k+1})};\theta_k)$. Comme ces lois ne sont pas explicites, nous les approchons en utilisant une méthode MCMC, et plus précisemment l'algorithme Hasting-Metropolis. Pour chaque distribution, plusieurs noyaux de transitions sont implémentés, associés à plusieurs lois candidates, de manière consécutive.
Après avoir défini le modèle étudié, nous expliquons brièvement l'algorithme SAEM sous une forme générale avant de détailler comment l'appliquer dans le cadre des modèles génétiques. Nous insistons sur les détails pratiques pour la mise en oeuvre de cette extension. Nous étudions ensuite deux applications sur des données réelles. Tout d'abord, nous étudions le cas d'un courbe de croissance pour le boeuf. Ces données proviennent d'une étude expérimentale dévellopée par l'INRA. Nous utilisons pour cela la fonction Brody, en considérant deux effets aléatoires corrélées, c'est à dire que l'on cherche à estimer les matrices de variance–covariance de dimension 2×2 des effets génétiques G et des effets environnementaux P en plus des effets fixes $\boldsymbol{\beta}$ et la variance des résidus σ^2.
Pour valider notre algorithme, nous avons réalisé une étude de simulation à partir de

ces données et des résultats obtenus avec SAEM. Nous avons donc simulé 400 jeux de données à partir de ces paramètres estimés et calculé la moyenne et la variance de ces 400 estimations obtenues avec notre algorithme ainsi que l'erreur quadratique moyenne observée pour chaque paramètre en montrant de très bons résultats. Finalement, à partir de ces données, une étude de comparaisons de modèles est proposée entre la courbe non linéaire Brody utilisée ici et les modèles de structures antédépendantes et les modèles de régression aléatoires.
Le deuxième exemple étudié dans ce chapitre porte sur une courbe de croissance chez le poulet à partir de données étudiées par Mignon-Grasteau et al. (2000). La fonction non linéaire de Gompertz a été utilisée ici pour modéliser ce problème en considérant trois effets aléatoires corrélés. Nous avons comparé nos résultats avec les résultats obtenus en utilisant WinBUGS pour une analyse Bayésienne mais nous avons rencontré divers problèmes de convergence. De plus, SAEM semble plus robuste vis–à–vis des points initiaux.

1.7 Une version "PX–SAEM"

Comme de nombreux algorithmes itératifs basés sur l'algorithme EM, SAEM peut être très lent à converger dans certains cas. Dans l'objectif d'améliorer la vitesse de convergence de l'algorithme SAEM, nous proposons d'adapter l'algorithme PX–EM, proposé par Liu et al. (1998) et qui permet d'accélérer l'algorithm EM de manière significative. Nous proposons dans le chapitre 3 une version *expansion de paramètres* de SAEM, que nous nommons PX–SAEM, pour accélérer l'algorithme SAEM.

L'idée centrale de l'algorithme PX–SAEM est de réaliser une extension de l'espace paramétrique à un ensemble plus grand que l'espace d'origine en incluant un paramètre de travail α. Le modèle d'origine a pour paramètre θ, alors que le modèle amplifié est paramétré en $\Theta = (\alpha,\theta)$. Pour pouvoir utiliser l'algorithme PX–SAEM deux conditions doivent être satisfaites: i) il existe α_0 tel qu'on récupère le modèle d'origine si $\alpha = \alpha_0$;
ii) il existe une fonction R de "réduction" à l'espace d'origine tel que $R(\Theta) = \theta$.
Si on se situe à l'itération k, on choisit $\alpha = \alpha_0$ et donc $\Theta_k = (\theta_k,\alpha = \alpha_0)$. Les étapes de Simulation et d'Approximation Stochastique de l'algorithme SAEM standard sont alors inchangées. A l'étape de Maximisation, on réactualise Θ en calculant Θ_{k+1} et on récupère les paramètres du modèle d'origine en appliquant la fonction de réduction R.

La version PX de l'algorithme SAEM est utile durant les premières itérations pour s'approcher d'un voisinage du maximum de vraisemblance. C'est pour cela que l'on propose un algorithme hybride, qui utilise PX–SAEM durant les premières itérations dans le

modèle amplifié et pour lesquelles les pas décroissants γ_k sont tous égaux à 1. On utilise l'algorithme SAEM standard pour le reste des itérations. La convergence presque sûre de l'algorithme vers ce maximum est donc assurée en utilisant une séquence décroissante (γ_k) sans expansion de paramètre.

Après avoir décrit les algorithmes EM et PX–EM ainsi que l'algorithme SAEM et sa version PX, nous présentons en détails l'application de notre nouvel algorithme dans le cadre des modèles mixtes linéaires et non linéaires. Deux applications sont ensuite analysées, la première étant un modèle linéaire mixte très simple qui nous permet de comparer les algorithmes EM et SAEM et leurs versions PX respectives, et le second exemple est un modèle mixte non linéaire étudiée par Concordet and Nunez (2002) et Kuhn and Lavielle (2005). A cause de la structure stochastique de l'algorithme SAEM, nous ne pouvons pas mesurer le gain de vitesse de convergence avec un critère de convergence, c'est pour cela que cette étude repose sur une analyse graphique de la convergence des paramètres estimés mais aussi sur l'évolution de l'estimation de la log–vraisemblance des observations en fonction des itérations.

Pour la première application étudiée, nous avons simulé des données à partir d'un modèle linéaire mixte très simple, en ne considérant qu'un seul effet aléatoire et deux effets fixes ($d = 1$ et $p = 2$ dans le modèle 1.2.1). Nous observons tout d'abord, comme prévu, qu'il n'y a pas de différences significatives entre les estimations obtenues avec EM et SAEM. Par contre, on voit clairement que les algorithmes PX de EM et SAEM convergent plus rapidement aux maxima de vraisemblance que les algorithmes standards.
Le deuxième exemple étudié est un modèle pharmacocinétique considérant deux effets aléatoires et deux effets fixes. L'étude de ce modèle effectuée par simulation a montré que l'algorithme SAEM était lent à converger voire incapable d'atteindre la convergence (sur 100 jeux de données étudiés, 20 d'entre eux ne convergent pas vers le maximum local en utilisant SAEM). L'algorithme PX–SAEM permet d'éviter sur ces jeux de données les deux problèmes, augmentant considérablement la vitesse de converge de l'algorithme SAEM standard, mais aussi, en évitant les maxima locaux puisqu'il converge vers le maximum global pour les 100 jeux de données étudiés.

1.8 Estimation REML pour les paramètres de variance dans les modèles mixtes non linéaires en utilisant l'algorithme SAEM

Ce travail est présenté au chapitre 4. Sa motivation est le fait qu'il est assez habituel d'obtenir des biais dans l'estimation des paramètres de variance dans les modèles mixtes

en utilisant la méthode du Maximum de Vraisemblance (ML). Jusqu'alors, ce problème a été largement étudié dans le cadre des modèles mixtes linéaires mais pas dans le domaine des modèles mixtes non linéaires. Un des atouts de l'algorithme SAEM est qu'il permet d'obtenir les estimateurs de Maximum de Vraisemblance dans ce type de modèle, c'est pourquoi il nous a semblé intéressant d'adapter un outil qui permet de réduire, dans le cas linéaire, le biais dans l'estimation des paramètres de variance.
Pour le modèle linéaire mixte, Dempster et al. (1977) et Laird and Ware (1982) ont montré que la méthode d'estimation du Maximum de Vraisemblance Restreinte ou Résiduelle (REML) peut être appliquée utilisant l'algorithme EM. La méthode REML consiste à maximiser la partie de la vraisemblance qui ne dépend que des paramètres de variance. Nous proposons donc dans ce travail d'utiliser la méthode REML avec l'algorithme SAEM dans le contexte des modèles mixtes non linéaires. Nous utilisons pour cela la version bayésienne de REML qui revient à intégrer les effets fixes (voir Harville (1974)). Cela équivaut à considérer les effets fixes comme aléatoires, en considérant une distribution a priori impropre. On se place donc dans le modèle suivant:

$$y_{ij} = g(t_{ij},\boldsymbol{\phi_i},\boldsymbol{\beta}) + \varepsilon_{ij}, \quad \text{pour } 1 \leq i \leq N, \quad 1 \leq j \leq n_i,$$

où:

$$\boldsymbol{\phi_i} = \boldsymbol{X_i\beta} + \boldsymbol{\eta_i} \quad \text{avec } \boldsymbol{\eta_i} \sim_{i.i.d.} \mathcal{N}(0,\Gamma).$$

On suppose que la distribution a priori $\pi(\cdot)$ de β est une distribution noninformative, c'est à dire que $\pi(\beta)$ est proportionnelle à une constante. Les données non observés sont donc maintenant $\boldsymbol{z} = (\boldsymbol{\eta},\boldsymbol{\beta})$ et les paramètres à estimer $\boldsymbol{\theta} = (\Gamma,\sigma^2)$. On peut donc appliquer l'algorithme SAEM–MCMC en considérant ce vecteur non observé $\boldsymbol{z}$. Pour réaliser l'étape de Simulation, on utilise un schéma de Gibbs, c'est à dire que l'on va simuler, à l'itération k, $\boldsymbol{\eta^{(k+1)}}$ à partir de $p(\cdot|\boldsymbol{y},\boldsymbol{\beta}^{(k)};\theta_k)$ et $\boldsymbol{\beta^{(k+1)}}$ à partir de $p(\cdot|\boldsymbol{y},\boldsymbol{\eta}^{(k+1)};\theta_k)$.

Notre étude sur données simulées montre que notre procédure d'estimation REML permet de corriger le biais des estimations des paramètres de variance dans les modèles mixtes non linéaires. De plus, notre travail montre que l'estimation REML est plus robuste que l'estimation ML dans des situations de données manquantes.

1.9 Application de l'algorithme SAEM dans les GLMM: modèle Probit

Les modèles pour données discrètes sont très importants dans plusieurs domaines de recherche, puisque les sujets sont souvent classés, ou répondent suivant une échelle binaire

ou ordinale. Dans ce travail, l'objectif est d'appliquer l'algorithme SAEM dans le contexte des Modèles Linéaires Mixtes Généralisés Mixtes (GLMM) et plus particulièrement, dans les cas de données binaires. Pour cela, nous avons décidé d'utiliser le modèle Probit.

Les données observées $\boldsymbol{y}$ sont binaires et l'on note p_{ij} la probabilité d'un événement positif, c'est à dire quand $y_{ij} = 1$, pour la j–ième observation de l'individu i, $1 \leq i \leq N$, $1 \leq j \leq n_i$. Le modèle Probit est définit comme il suit:

$$\begin{aligned} y_{ij} &\sim \text{Ber}(p_{ij}), \quad \text{avec} && (1.9.13)\\ p_{ij} &= P(y_{ij} = 1) && (1.9.14)\\ &= \Phi(\boldsymbol{X_{ij}}\boldsymbol{\beta} + \boldsymbol{Z_{ij}}\boldsymbol{\eta_i}) \end{aligned}$$

où $\boldsymbol{\eta_i} \in \mathbb{R}^d$ suit une loi gaussienne $\mathcal{N}(0,\boldsymbol{\Gamma})$ et $\Phi(x)$ représente la fonction de répartition de la loi Normale de moyenne x et variance 1. $\boldsymbol{X_{ij}}$ et $\boldsymbol{Z_{ij}}$ sont des matrices connues de dimension respectives $1 \times p$ et $1 \times d$.

Il est aussi possible d'écrire ce modèle en utilisant une variable latente ω_{ij}:

$$\begin{aligned} y_{ij} &= sign(\omega_{ij}) = \begin{cases} 1 \text{ si } \omega_{si} > 0 \\ 0 \text{ si } \omega_{ij} \leq 0 \end{cases}, \text{ avec} && (1.9.15)\\ \omega_{ij} &= \boldsymbol{X_{ij}}\boldsymbol{\beta} + \boldsymbol{Z_{ij}}\boldsymbol{\eta_i} + \varepsilon_{ij}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq n_i && (1.9.16) \end{aligned}$$

où $\varepsilon_{ij} \sim \mathcal{N}(0,1)$. On suppose que $(\boldsymbol{\eta_i})$ et $(\boldsymbol{\varepsilon_i})$ sont mutuellement indépendants.

L'algorithme SAEM est donc applicable si on considère comme données non observées $\boldsymbol{z} = (\boldsymbol{\omega},\boldsymbol{\eta})$, où $\boldsymbol{\omega} = (\omega_{ij}, 1 \leq i \leq N, 1 \leq j \leq n_i)$ et $\boldsymbol{\eta} = (\boldsymbol{\eta_i}, 1 \leq i \leq N)$, et $\boldsymbol{y} = (y_{ij}, 1 \leq i \leq N, 1 \leq j \leq n_i)$ comme étant les données observées. Les paramètres à estimer sont donc $\boldsymbol{\theta} = (\boldsymbol{\beta},\boldsymbol{\Gamma})$. L'étape de Simulation est assez simple puisque les lois conditionnelles $\boldsymbol{\omega}|\boldsymbol{\eta},\boldsymbol{y};\boldsymbol{\theta}$ et $\boldsymbol{\eta}|\boldsymbol{\omega},\boldsymbol{y};\boldsymbol{\theta}$ sont connues.

L'algorithme SAEM peut aussi présenter des problèmes de vitesse de convergence dans ce domaine des GLMM, c'est pourquoi nous adaptons l'algorithme PX–SAEM à ce type de modèle. Nous présentons aussi une procédure pour obtenir les estimateurs REML des paramètres de variance dans le cadre des GLMM utilisant l'algorithme SAEM.

Deux applications sont étudiées pour illustrer toutes ces méthodes. Nous analysons tout d'abord un problème qui porte sur des données réelles de patients atteints d'épilepsie. Les estimations obtenues avec SAEM sont très proches de celles obtenues avec la procédure NLMIXED du logiciel statistique SAS. De plus, sur ces données, l'algorithme PX–SAEM améliore considérablement la vitesse de convergence de l'algorithme SAEM.

Le second exemple étudié est une étude basée sur des données de patients atteints de schizophrénie. Il s'agit d'un exemple beaucoup plus complexe puisque les données sont non équilibrées et nous avons un grand nombre d'observations, 437 patients. De plus, le modèle utilisé ici considère deux effets aléatoires corrélés. Sur cet exemple, nous avons calculé les estimations REML des paramètres de variance obtenant des estimations plus grandes en comparaison des estimations ML. Nous avons aussi comparé nos résultats avec ceux obtenus avec la procédure NLMIXED du logiciel statistique SAS montrant que l'algorithme SAEM est beaucoup plus stable vis–à–vis de l'initialisation.

En guise de conclusion, nous discutons des nombreuses extensions possibles de l'algorithme SAEM dans le cadre de GLMM.

Chapitre 2

Genetic analysis of growth curves using the SAEM algorithm

Summary

The analysis of nonlinear function-valued characters is very important in genetic studies, especially for growth traits of agricultural and laboratory species. Inference in nonlinear mixed effects models is, however, quite complex and is usually based on likelihood approximations or Bayesian methods. The aim of this work is to present an efficient stochastic EM procedure, namely the SAEM algorithm, which is much faster to converge than the classical Monte Carlo EM algorithm and Bayesian estimation procedures, does not require specification of prior distributions and is quite robust to the choice of starting values. The key idea is to recycle the simulated values from one iteration to the next in the EM algorithm, which considerably accelerates the convergence. A simulation study is presented which confirms the advantages of this estimation procedure in the case of a genetic analysis. The SAEM algorithm was applied to real data sets on growth measurements in beef cattle and in chicken. The proposed estimation procedure, as the classical Monte Carlo EM algorithm, provides significance tests on the parameters and likelihood based model comparison criteria to compare the nonlinear models with other longitudinal methods.

This chapter corresponds, with more details, to an article which is a joint work with Florence Jaffrézic, Marc Lavielle and Jean-Louis Foulley accepted to *Genetics Selection Evolution.*

Contents

2.1 Introduction

Many traits of interest in genetic studies are function-valued characters, i.e. they change in a continuous manner over time or some other independent continuous variable. Focus will be in this study on nonlinear functions applied to growth traits. They are of interest for many agricultural and laboratory species such as rabbits (Blasco et al., 2003), chicken (Mignon-Grasteau et al., 2000), pigs (Huisman et al., 2002), cattle (Jaffrezic et al., 2004), mice (Atchley and Zhu, 1997) and trees (Ma et al., 2002).

Various methodologies have been proposed to analyze such longitudinal data, including random coefficient models (Diggle et al., 1994), which model individual deviations with polynomial functions of time, and structured antedependence models (Nunez-Anton and Zimmerman, 2000; Jaffrezic et al., 2003), which consider that the observation at time t is a function of the previous observations. These models are in the linear mixed model framework and can be implemented in traditional mixed model software.

A different approach for function-valued characters, especially growth traits, is to use a parametric nonlinear function of time, with a few interpretable parameters, that are decomposed into a genetic and an environmental component. For instance, the Gompertz curve has proved suitable for modelling growth curves in rabbits (Blasco et al., 2003) and chicken (Mignon-Grasteau et al., 2000). It has three parameters that have an interesting

biological interpretation in terms of adult body weight and maturation rate. This modelling is similar in spirit to the random regression approach, but it overcomes the drawbacks encountered with the use of polynomimal functions. This nonlinear modelling of growth curves has also been used in QTL detection by Ma et al. (2002).

Estimation procedures for these nonlinear mixed effects models are, however, much more complex, and require the use of stochastic estimation procedures. Some authors have used the Gibbs sampling for Bayesian estimations (Blasco et al., 2003). These Bayesian methods do, however, have a few drawbacks such as the choice of prior distributions, the computing time, the check of convergence and inference on the estimated parameters (significance tests, etc.).

On the other hand, McCulloch (1997) proposed using a hybrid algorithm combining a Markov Chain Monte Carlo EM algorithm - MCEM (Wei and Tanner, 1990a) and a Markov Chain Monte Carlo (MCMC) integration and maximization of the likelihood - MCMLE (Geyer, 1994). Indeed, the MCEM algorithm converges quickly to the neighbourhood of the parameter estimates, but shows a great deal of variability within this neighbourhood. And, it requires a considerable increase in the number of MCMC draws and the number of EM iterations to make the procedure accurate (Booth and Hobert, 1999). On the other hand, the MCMLE algorithm provides accurate estimates as well as all the elements required for parameter testing and model comparisons. It is, however, very computationally expensive and requires a reference point in the parameter space close to the actual MLE (Pletcher and Jaffrézic, 2002).

The aim of this work is to present an extension of the stochastic approximation EM algorithm (SAEM) proposed in the statistical literature (Kuhn and Lavielle, 2005) and to apply it to the genetic analysis of growth curves. This methodology combines the strength of the two aforementioned algorithms. As with the MCEM algorithm it is quite robust to starting values, but has much faster convergence to the maximum likelihood estimates, thanks to a smoothing parameter. It also provides the likelihood value and confidence intervals for all the estimated parameters, and therefore permits the use of classical significance tests and likelihood based model comparison criteria.

Section 2.2 describes the genetic model used here. In Section 2.3 we described the SAEM algorithm for genetic studies. A simulation study will be presented in Section 2.4 to check the properties of this algorithm in genetic studies, and an application to growth data analysis in beef cattle and in chicken will be presented.

TAB. 2.1 – *Equations of growth curves*

Curve	Model for the weight at age t
Brody	$A - B\exp(-Kt)$
Gompertz	$A\exp(-B\exp(-Kt))$
Richards	$A(1+\delta B\exp(-Kt))^m$ $\delta = 1$ if $m < 0; \delta = -1$ if $m > 0$
Logistic	$A(1+\exp(-Kt))^{-1}$
Janoschek	$A - (A - W_0)\exp(-Rt^p)$ $(p \neq 0)$

A: asymptotic weight

K: maturation rate

W_0: birth weight

B: constant value generally linked with W_0

m,p: shape parameter

R: speed of growth parameter

2.2 The genetic models

Many phenotypes of an individual animal, such as for example body weight, change with age. There is evidence that changes in performance of animals with age are influenced by genetic factors. These genetic parameters reflect to what extent and how genetic changes in performance patterns over time can be achieved by selection. It was showed that phenotypic changes with age could be represented as a function of time. Traditionally traits that are measured in time are analyzed with an multitrait model, defining the phenotypic values at distinct ages as different traits.

Growth curves can describe the entire growth process in terms of a few parameters having a biological interpretation. Many mathematical functions were used to describe the growth curves, but typically they are fitted by nonlinear regression. Table 2.1 resumes the most popular.

We consider in this work the following hierarchical model. For the animal i, we observed the weight y_{ij} at age t_{ij}, with $i = 1, \cdots, N$ and $j = 1, \cdots, n_i$. A first stage describes the growth curve, using one of the function described in Table 2.1. A second stage describes the individual variability, considering a linear model that includes environmental and

genetics effects. In a general form, the model can be written as:

$$y_{ij} = g(t_{ij},\phi_i) + h(t_{ij},\phi_i)\varepsilon_{ij} \tag{2.2.1}$$

where $y_{ij} \in \mathbb{R}$ denotes the jth observation of the subject i, at time t_{ij}. The within–group errors (ε_{ij}) are supposed to be *i.i.d.* Gaussian random variables with mean zero and unknown variance σ^2.

The model will be nonlinear when g or h are nonlinear functions of the individual random parameters $\phi_i \in \mathbb{R}^d$. In this context, g is a function defined in Table 2.1 and in the case where g is the Brody function, for instance, and h is equal to 1, the model reduces to:

$$y_{ij} = A_i - B_i\, e^{-K_i t_{ij}} + \varepsilon_{ij} \tag{2.2.2}$$

where t_{ij} is the time of measurement. The individual vector of parameters is $\phi_i = (A_i,B_i,K_i)$, that are biologically interpretable.

In the case of a genetic analysis, for an animal model, vector ϕ_i for individual i is decomposed as follows:

$$\phi_i = X_i\beta + Z_i u + \eta_i \tag{2.2.3}$$

where β are the fixed effects influencing the curve parameters (A_i,B_i,K_i), u are the genetic effects and η_i are the permanent environmental effects. Matrices X_i and Z_i are known incidence matrices. It is assumed that u of dimension $dN_a \times 1$ is normally distributed: $u \sim \mathcal{N}(0,A \otimes G)$, where matrix G is of dimension $d \times d$ (for example 3×3 in the case of the Brody function) and represents the genetic covariance matrix between the curve parameters (A_i,B_i,K_i), and matrix A is the known genetic relationship matrix. The environmental vector η_i is also assumed normally distributed, with mean zero and covariance matrix P, of dimension $d \times d$, which represents the environmental covariance matrix between the curve parameters. Let θ be the vector of parameters to be estimated: $\theta = (\beta,G,P,\sigma^2)$.

2.3 The SAEM algorithm for genetic studies

2.3.1 Description of the algorithm

In the EM framework, we consider that the missing data is $z = (\phi,u)$. We propose to use the Stochastic Approximation version of EM algorithm, introduced by Delyon et al. (1999) and generalized for the nonlinear mixed effects models by Kuhn and Lavielle

(2005), to obtain the Maximum Likelihood estimates in the context of genetic studies.

The general idea of SAEM algorithm is to replace the Expectation phase of the EM algorithm, i.e. the calculation of the conditional expectation of the likelihood of the complete data, by a simulation step and a stochastic approximation. The Maximization step is unchanged.

In this context, the k-th step of SAEM is as follows:

- *Simulation–Step*: generate m realizations $\boldsymbol{z}^{(k+1,l)}$ $(1 \leq l \leq m)$ from $p(\boldsymbol{z}|\boldsymbol{y};\boldsymbol{\theta_k})$.

- *Stochastic approximation–Step*: update $Q_k(\boldsymbol{\theta})$ according to

$$Q_{k+1}(\boldsymbol{\theta}) = Q_k(\boldsymbol{\theta}) + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\log f(y,\boldsymbol{z}^{(k+1,l)};\boldsymbol{\theta}) - Q_k(\boldsymbol{\theta})\right), \tag{2.3.4}$$

  where (γ_k) is a sequence of positive step sizes decreasing to 0.

- *Maximization–Step*: compute $\boldsymbol{\theta_{k+1}}$ which maximizes $Q_{k+1}(\boldsymbol{\theta})$.

When the simulation of the sequences $\boldsymbol{z}^{(k+1,l)}$ cannot be directly perform, as it is the case in this model, Kuhn and Lavielle (2005) propose to combine this algorithm with a Markov Chain Monte Carlo (MCMC) procedure. The Simulation–Step becomes:

- *Simulation–step*: using $\boldsymbol{z}^{(k)}$, draw $\boldsymbol{z}^{(k+1)}$ from transition probability $\Pi_{\Theta_k}(\boldsymbol{z}^{(k)},\cdot)$.

Precise results of convergence of SAEM are presented in Delyon et al. (1999). These results were extended by Kuhn and Lavielle (2005) when a MCMC procedure is used.

2.3.2 Application to the genetic model

If we assume that the model 2.2.1 belongs to the exponential family, then the complete log–likelihood can be written as:

$$\log p(\boldsymbol{y},\boldsymbol{\phi},\boldsymbol{u};\boldsymbol{\theta}) = -\Psi(\boldsymbol{\theta}) + \langle S(\boldsymbol{y},\boldsymbol{z}),\xi(\boldsymbol{\theta})\rangle$$

where $S(\boldsymbol{y},\boldsymbol{z})$ is the sufficient statistics of the complete–data model.

Then the Stochastic approximation–step of the SAEM algorithm reduces to compute:

$$s_{k+1} = s_k + \gamma_{k+1}\left(S(\boldsymbol{y},\boldsymbol{z}^{(k+1)}) - s_k\right). \tag{2.3.5}$$

The Maximization–step is the same since we compute $\boldsymbol{\theta_{k+1}}$ which maximizes $Q_{k+1}(\boldsymbol{\theta})$.

In this context, the log–likelihood of the complete data $\log p(\boldsymbol{y},\boldsymbol{\phi},\boldsymbol{u})$ can therefore be decomposed as:

$$\begin{aligned}
\log p(\boldsymbol{y},\boldsymbol{\phi},\boldsymbol{u};\boldsymbol{\theta}) &= \log p(\boldsymbol{y}|\boldsymbol{\phi},\boldsymbol{u};\boldsymbol{\theta}) + \log p(\boldsymbol{\phi}|\boldsymbol{u};\boldsymbol{\theta}) + \log p(\boldsymbol{u};\boldsymbol{\theta}) \qquad (2.3.6)\\
&= -\frac{N_{tot}+dN+N_a}{2}\log(2\pi) - \frac{N_{tot}}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{n_i}(y_{ij}-g(t_{ij},\boldsymbol{\phi_i}))^2\\
&\quad -\frac{N}{2}\log(|\boldsymbol{P}|) - \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{\phi_i}-\boldsymbol{Z_i u}-\boldsymbol{X_i\beta})'\boldsymbol{P}^{-1}(\boldsymbol{\phi_i}-\boldsymbol{Z_i u}-\boldsymbol{X_i\beta})\\
&\quad -\frac{N_a}{2}\log(|\boldsymbol{\Gamma}|) - \frac{1}{2}\boldsymbol{u}'\boldsymbol{\Gamma}^{-1}\boldsymbol{u}
\end{aligned}$$

where $N_{tot} = \sum_{i=1}^{N} n_i$ is the total number of observations, d is the dimension of vector $\boldsymbol{\phi_i}$, for all individuals i ($d = 3$ for a Brody function for example: $\boldsymbol{\phi_i} = (A_i,B_i,K_i)$), and N_a is the number of animals in the relationship matrix. Let $\boldsymbol{\Gamma} = \boldsymbol{A} \otimes \boldsymbol{G}$ be the genetic covariance matrix.

Then, at the iteration k, the Stochastic approximation–step updates the sufficient

statistics of the complete–data model as follows:

$$s_1^{(k+1)} = s_1^{(k)} + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\sum_{i=1}^{N} \boldsymbol{X_i}'\boldsymbol{P}^{-1}\left(\boldsymbol{\phi_i}^{(k+1,l)} - \boldsymbol{Z_i u}^{(k+1,l)}\right) - s_1^{(k)}\right),$$

$$s_{2(r,t)}^{(k+1)} = s_{2(r,t)}^{(k)} + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m} \boldsymbol{u_r}^{(k+1,l)\prime}A^{-1}\boldsymbol{u_t}^{(k+1,l)} - s_{2(r,t)}^{(k)}\right) \quad \text{for } r,t = 1,\cdots,d$$

where $\boldsymbol{u_r}$ and $\boldsymbol{u_t}$ are of dimension $(N_a \times 1)$,

$$s_{3(r,t)}^{(k+1)} = s_{3(r,t)}^{(k)} + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\sum_{i=1}^{N}\left(\boldsymbol{\phi_{i,r}}^{(k+1)} - \boldsymbol{Z_i u_r}^{(k+1)} - \boldsymbol{X_i\beta_r}^{(k+1)}\right)'\right.$$

$$\left.\times\left(\boldsymbol{\phi_{i,t}}^{(k+1)} - \boldsymbol{Z_i u_t}^{(k+1)} - \boldsymbol{X_i\beta_t}^{(k+1)}\right) - s_{3(r,t)}^{(k)}\right),$$

$$s_4^{(k+1)} = s_4^{(k)} + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\sum_{i=1}^{N}\sum_{j=1}^{n_i}(y_{ij} - f(t_{ij},\boldsymbol{\phi_i}^{(k+1)}))^2 - s_4^{(k)}\right).$$

The Maximization–step can be written according to the sufficient statistics:

$$\begin{aligned}
\boldsymbol{\beta}_{k+1} &= \left(\sum_{i=1}^{N} X_i'\boldsymbol{P}^{-1}X_i\right)^{-1} s_1^{(k+1)} \\
\boldsymbol{G}_{(r,t)}^{k+1} &= \frac{s_{2(r,t)}^{(k+1)}}{N_a} \\
\boldsymbol{P}_{(r,t)}^{k+1} &= \frac{s_{3(r,t)}^{(k+1)}}{N} \\
\sigma_{k+1}^2 &= \frac{s_4^{(k+1)}}{N_{tot}}.
\end{aligned}$$

In practice, the non observed data $\boldsymbol{z} = (\boldsymbol{\phi},\boldsymbol{u})$ can be simulated using a Gibbs scheme. At iteration k, we draw $\boldsymbol{\phi}^{(k+1)}$ from the conditional distribution $p(\cdot|\boldsymbol{y},\boldsymbol{u}^{(k)};\boldsymbol{\theta_k})$ and then $\boldsymbol{u}^{(k+1)}$ from $p(\cdot|\boldsymbol{y},\boldsymbol{\phi}^{(k+1)};\boldsymbol{\theta_k})$. Furthermore, we use the Hasting-Metropolis algorithm to approximate these conditional distributions. For each distribution, several transition kernels are implemented, associated to different proposals can be successively used.

Parameter γ_k is a crucial parameter in this estimating procedure. It performs a smoothing of the calculated likelihood values from one iteration to the other and therefore

considerably accelerates convergence compared to other MCMC estimation procedures. In practice, this smoothing parameter is defined as follows. During the first K iterations, γ_k=1, i.e. there is no smoothing performed and the algorithm is equivalent to an MCEM algorithm (Meng and Rubin, 1993b). McCulloch (1997) showed that this algorithm converged very rapidly towards a neighbourhood of the ML estimates but then continued showing a great deal of variation. Therefore, from iteration $(K+1)$ the smoothing starts in order to stabilize the estimates and converge more rapidly towards the actual ML estimates (Kuhn and Lavielle, 2005). Parameter γ_k is a sequence of stepsizes within the interval [0,1]. It is recommended (Kuhn and Lavielle, 2005) to take $\gamma_k = (k - K)^{-1}$ for $k \geq (K + 1)$. The choice of the iteration number K can depend on the number of simulations performed at each iteration. To ensure the algorithm has already converged into a neighbourhood of the MLEs before the smoothing starts, it is recommended to use this algorithm with several different starting values.

An advantage of the stochastic EM approach is that it remains in the classical maximum likelihood framework. It therefore allows the calculation of the likelihood value of the model using Importance Sampling and the calculation of the SE of the parameters using Louis's missing information principle (Louis, 1982) as presented by Lavielle (2005). This enables significance tests on the parameters (fixed effects and variance-covariance components) and also enables model comparisons using classical criteria such as likelihood ratio tests, AIC or BIC criteria.

A Matlab program is available for genetic analyses using the SAEM algorithm.

2.4 Examples

2.4.1 Growth curve analysis in beef cattle

Data analyzed in this study came from an INRA experimental Charolais herd (Mialon et al., 2001). The data set comprised body weight records for 560 cows, born over an 11 year period (from 1988 to 1998), from 60 sires and 369 dams. Data were collected monthly from 1998 to 2003, but only 10 measurements from each animal were included being at around 0, 112, 224, 364, 540, 720, 900, 1260, 1620 and 1980 days. Although the same ages were considered for each animal, they were unequally spaced and some records were missing.

A Brody function was used to analyze these data and a sire model was considered.

The model can be written as:

$$y_{ij} = A_i - B_i e^{-K_i t_j} + \epsilon_{ij} \tag{2.4.7}$$

where y_{ij} is the body weight measurement for individual i at time t_j (t_j corresponds to the ages of measurement divided by 100000). The two individual parameters of this nonlinear function: A_i and K_i have an interesting biological interpretation. In fact, A_i represents the adult body weight for individual i and K_i is its maturation rate. Due to identifiability problems, a reparametrization was used here for the classical B_i parameter of the Brody function such that $B_i = A_i - W0_i$, where $W0_i$ is the observed birth weight. The residual term ϵ_{ij} was assumed normally distributed with mean zero and constant variance σ^2. Parameters A_i and K_i are also assumed normally distributed and are decomposed using a sire model, as a special case of the animal model presented in the methodology section above (equation 2.2.3).

Analysis with the SAEM algorithm

As shown in Figure 2.1, the Brody function is very appropriate to model the growth curves in beef cattle. Estimates obtained for each of the parameters are given in Table 2.2. As expected, the genetic correlation between parameters A and K was quite high (-0.80). It still is, however, different from 1 which gives the possibility for a genetic selection for high growth rate while keeping a reasonable adult body weight, which is the goal of beef cattle breeders.

In order to check the accuracy of the SAEM estimates, we simulated 400 data sets with these parameter values, and Table 2.3 provides the mean, variance and relative mean square error (RMSE) for each of these parameters over the 400 data sets. Estimations for all the simulated data sets were performed with 700 iterations, with the smoothing parameter starting after 400 iterations, 5 chains and 8 simulations per chain at each iteration (which corresponds to a total of 28000 MC samples).

Analysis of these 400 data sets was performed using different starting values. The SAEM algorithm was found to be robust to the choice of starting values for the variance parameters. On the other hand, starting values for the fixed effects should be quite close to the real parameter values. Good initial values for the fixed effects can easily be obtained with the NLIN procedure of SAS, for example. The algorithm was found to converge better when initial values for the variance components were larger than the expected ones.

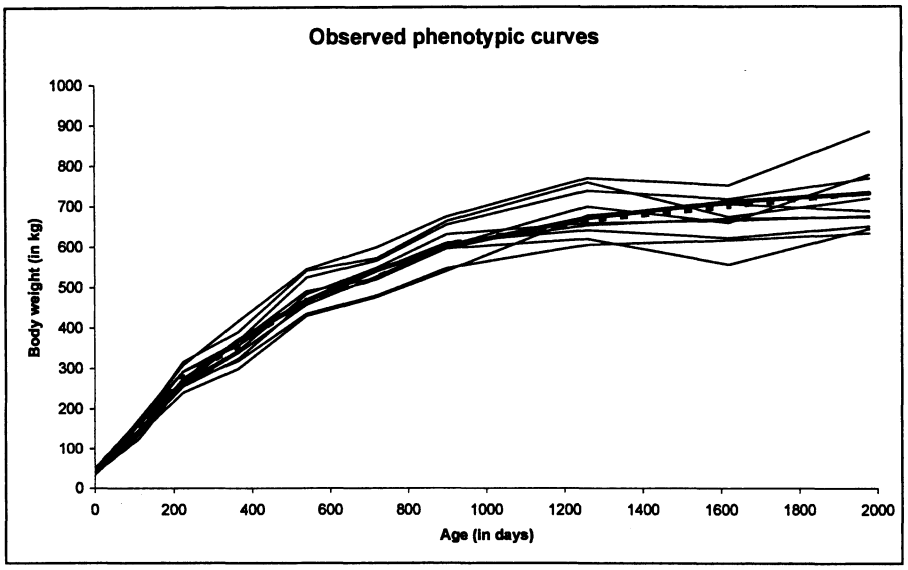
TAB. 2.2 – *Estimated genetic sire variances and correlation (VarG, CorrG) for the curve parameters A and K and permanent environmental variances and correlation (VarE, CorrE) for A and K with the SAEM algorithm for the beef cattle growth data using a Brody function. (In brackets are the SE of the parameters.)*

Fixed effects			
μ_A	761 (4.08)	μ_K	165 (1.32)
Variance components			
VarG_A	1270 (411)	VarE_A	4190 (309)
VarG_K	54.8 (21.9)	VarE_K	518 (43.2)
CorrG_{AK}	-0.80	CorrE_{AK}	-0.71
Residual variance		687 (14.4)	

TAB. 2.3 – *Estimated genetic and environmental parameters with the SAEM algorithm for 400 simulated data sets with a sire model and the Brody function ($\boldsymbol{\theta_0}$ represents the starting values).*

	μ_A	μ_K	VarG_A	VarG_K	CorrG_{AK}	VarE_A	VarE_K	CorrE_{AK}	σ^2
Simulated	760	165	1300	60	-0.80	4200	520	-0.72	690
$\boldsymbol{\theta_0}$	800	200	15000	6000	0.0	15000	6000	0.0	12869
Mean	760.2	164.9	1256.6	62.3	-0.80	4214.5	512.1	-0.72	690.4
Variance	31.9	2.26	103410	621.5	0.0121	106040	2275.1	0.0008	248.2
RMSE%	0.74	0.91	25.0	41.7	13.7	7.8	9.3	3.9	2.3

FIG. 2.1 – *Phenotypic curves*



Comparison with other nonlinear estimation procedures on these simulated data sets is difficult due to the computing time required by Bayesian analyses and the difficulty for approximated methods such as FOCE - First Order Conditional Estimation (Lindstrom and Bates, 1990) to analyze any sampled data set arising from a simulation study. In addition, most softwares based on the Gaussian quadrature such as SAS NLMIXED do not allow a random structure as complex as this one. Concerning the computing time, the phenotypic analysis of the real data set was performed with both the SAEM algorithm and the Gibbs sampling using the winBUGS program (Spiegelhalter et al., 2004). The SAEM algorithm converged and provided accurate parameter estimations in less than 4 minutes (for 700 iterations, 5 chains and 8 simulations per chain), whereas the Gibbs sampling required at least 50000 iterations, which took about 30 minutes to run.

Model comparisons

A previous study showed that the structured antedependence (SAD) models performed well to analyze this growth pattern compared to the classical random regression (RR) models (Jaffrezic et al., 2004). The aim is now to compare these models and the proposed nonlinear approach. Model comparison was based on the likelihood values and the BIC criterion, which was calculated using the following formula: BIC $= -2$ LogL $+$ n_c Log(N) where -2 LogL is minus twice the log-likelihood value, n_c is the number of covariance parameters in the model and N is the total number of observations. Notice that N in the previous formula has to be replaced by (N-p) (where p is the number of fixed effects, also equal to rank($\mathbf{X}$)) in the case of REML estimation.

In order to compare the different methodologies, the same mean curve was used as fixed effects, i.e. the Brody curve presented above ($f(t) = a - b\exp(-kt)$). For the SAD and RR models, as the nonlinear parameter k could not be estimated with ASREML (Gilmour et al., 2004), the value obtained with the SAEM algorithm was used. The aim was to compare the flexibility of the three approaches to model the covariance structure. To do so, the variances and correlations were calculated at each of the 10 ages with the three methods (SAD, RR, Brody). As no analytical form is available for a nonlinear model to calculate the variance and correlation functions, they were calculated by simulations. In order to have a 'reference' model, this analysis was performed in the phenotypic case, and the three estimated covariance structures were compared to a completely unstructured model.

To make sure the likelihood values were comparable, the 10 by 10 phenotypic covariance matrix was calculated with the parameters obtained with each of the models and

fixed in ASREML (for US, SAD, RR and Brody) to obtain the likelihood values.

Table 2.4 gives the likelihood values and BIC criterion for the different models ('US': the completely unstructured model with a 10 by 10 estimated covariance matrix; 'SAD2-quad-const': second order structured antedependence model with a quadratic first order antedependence parameter and constant second order; 'RR cubic': random regression model based on a polynomial of order 3).

The unstructured model (US) was found here to have the smallest BIC value and is considered as the 'reference' model. It was found that although the nonlinear shape of the curve is very appropriate to model the phenotypic growth phenomenon, it is less flexible than the structured antedependence and even the cubic random regression model to fit the covariance structure. In fact, as shown in Figure 2.2, the Brody model did not fit the correlation pattern very well; the estimated correlations were underestimated at early ages and slightly overestimated at late ages. Similarly, the phenotypic variance shown in Figure 2.3 was overestimated at early ages and underestimated at late ages. On the other hand, although the likelihood value and BIC criterion were higher for the cubic random regression model than for the Brody function, Figure 2.2 shows that the use of the nonlinear Brody function avoided the main drawbacks of the random regression models based on polynomial functions, which are the border effects.

TAB. 2.4 – *Likelihood values and BIC criterion for the phenotypic analysis (the smaller the values are the better the model is). 'Nb Par Cov' is the number of parameters in the covariance structure. To make the model comparisons easier a constant ($c = -40000$) was added to all the likelihood values.*

Model	Nb Par Cov	-2 Log L	BIC
US	55	901.6	1374.8
SAD2 quad-const	7	1592.2	1652.4
RR cubic	11	2732.2	2826.8
BRODY	4	3382.4	3416.8

The Brody model also requires the estimation of only very few parameters and allows the direct prediction of individual genetic values for the adult body weight and the maturation rate, which is quite difficult to define with other longitudinal models.

FIG. 2.2 – *Estimated phenotypic correlation functions obtained with the unstructured (US), SAD, RR and Brody models presented in Table 2.4.*

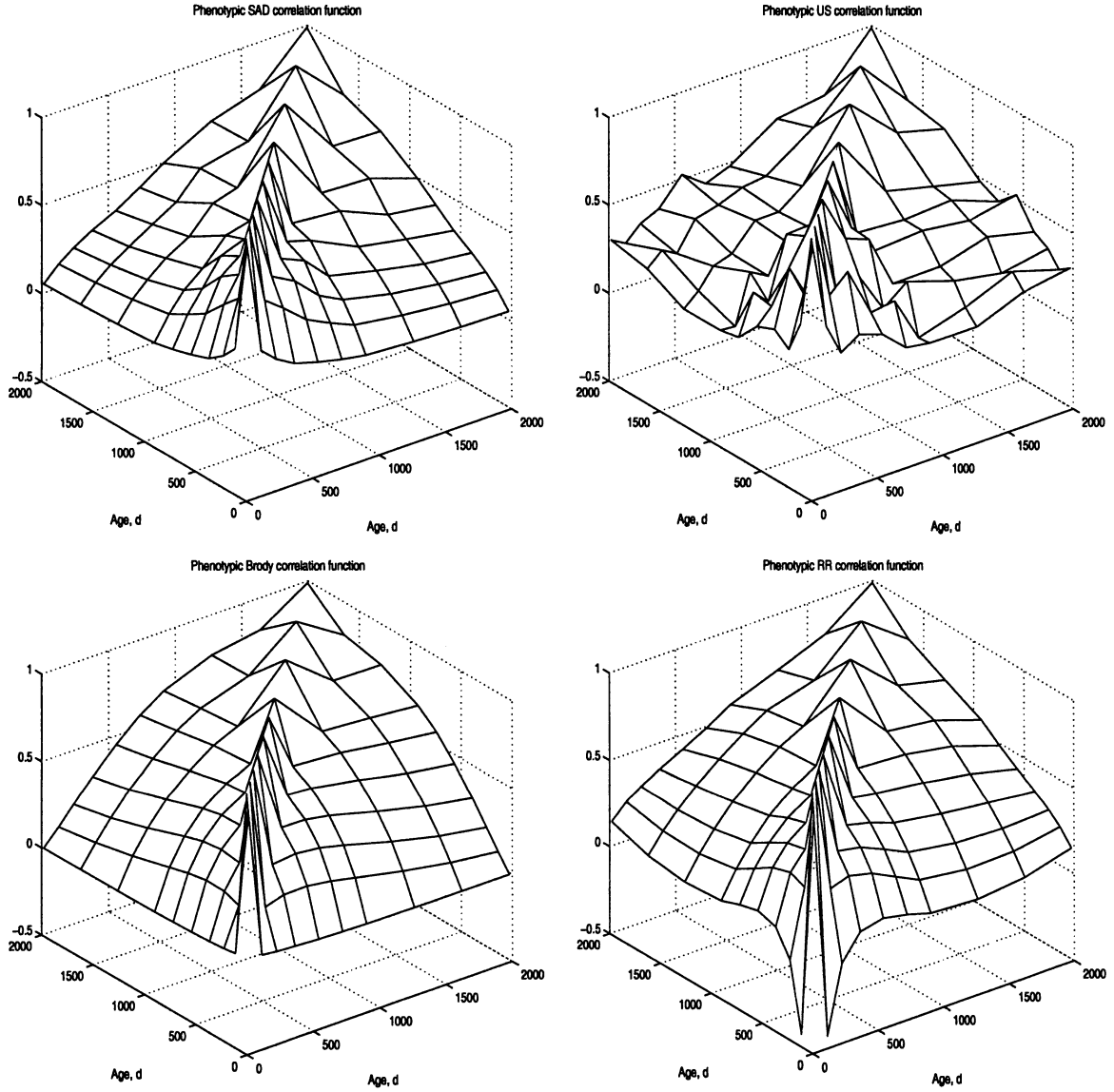
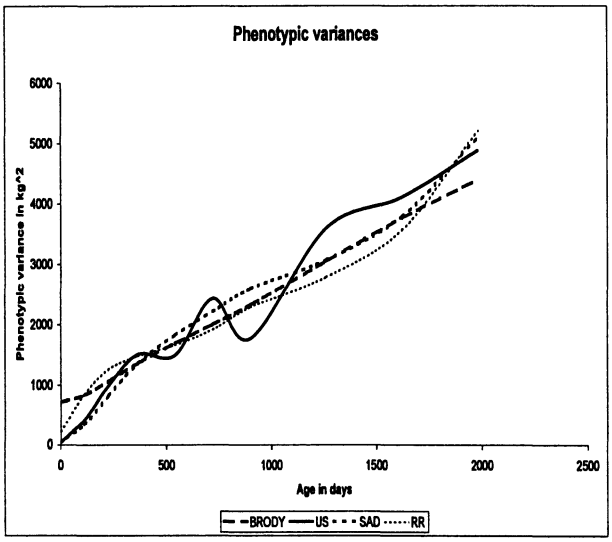


FIG. 2.3 – *Estimated phenotypic variance functions obtained with the unstructured (US), SAD, RR and Brody models presented in Table 2.4.*



2.4.2 Growth curve analysis in chicken

This data set corresponds to the last generation of selection from the experiment presented by Mignon-Grasteau et al. (2000). Data originated from a selection experiment on the form of the growth curve initiated by F. Ricard in 1960 on meat-type chickens. Line X+- was selected for high juvenile body weight at 8 weeks and low adult body weight at 36 weeks. In contrast, line X-+ was selected for low juvenile body weight and high adult body weight. In line X++, chicken were selected for high body weights at both ages and, in the opposite line, X- -, they were selected for low body weights at both ages. Line X00 was an unselected control line. The data set comprised in total 265 chicken, from 71 sires, and about 12 measurements for each animal at age 0, 4, 6, 8, 12, 16, 20, 24, 28, 32, 36 and 40 weeks. Only chicken with more than 5 measurements were included in the analyses.

The Gompertz function was used and the model can be written as follows:

$$y_{ij} = A_i \exp(-B_i \exp(-K_i t_j)) + \epsilon_{ij} \tag{2.4.8}$$

where A_i is the asymptotic body weight of chicken i, i.e. the weight at an infinite age. Parameter B_i is equal to $\ln(A_i/W_{i0})$ where W_{i0} is the estimated hatching weight of chicken i. Parameter K_i corresponds to the maturation rate, i.e. the rate at which the animal approaches its asymptotic weight. In this equation, the times t_j correspond to the ages of measurement listed above divided by 100. A sire model was used for each of these three parameters, and the different lines were fitted as fixed effects. As before, the three parameters of the curve were assumed normally distributed and correlated. The residuals ϵ_{ij} were also assumed normally distributed with mean zero and constant variance σ^2.

As mentioned in the methodology section, the SAEM approach allows to perform significance tests on the parameters. Using a likelihood ratio test, it was found that the environmental covariance between parameters A and B of the Gompertz function was not significant. It was therefore set to zero.

On the other hand, it was found that the line effects were all significantly different for the three parameters of the curve. As expected and as shown in Table 2.5, the mean effect for parameter A, i.e. the asymptotic body weight, was found the highest for lines X++ and X-+, and the lowest for lines X- - and X+-. On the other hand, the maturation rate (parameter K) was found the lowest for line X-+ and the highest for line X+-.

Table 2.6 provides the estimated genetic and environmental variance and correlation parameters. As they were calculated only on the last generation of selection, they were found to be different from the results obtained by Mignon-Grasteau et al. (2000).

TAB. 2.5 – *Estimated fixed effects with the SAEM algorithm for the chicken growth data using a sire model and the Gompertz function. (In brackets are the SE of the parameters.)*

	Line X-+	Line X+-	Line X++	Line X- -	Lin X00
μ_A	3070 (49.4)	1960 (47.3)	3110 (44.0)	1750 (41.0)	2350 (30.2)
μ_B	4.73 (0.0971)	3.36 (0.203)	4.36 (0.13)	4.39 (0.0502)	3.72 (0.0573)
μ_K	12.7 (0.454)	16.7 (0.811)	16.5 (0.586)	15.4 (0.332)	14.8 (0.29)

TAB. 2.6 – *Estimated genetic and environmental variances and correlations obtained with the SAEM algorithm for the chicken growth data using a sire model and the Gompertz function. On the diagonal are the variances and off-diagonal are the correlations. (In brackets are the SE of the parameters).*

Genetic components			
A	6220 (6960)	-0.12	-0.36
B		0.0428 (0.00866)	0.96
K			1.28 (0.128)
Environmental components			
A	49000 (7450)	0	-0.35
B		0.0194 (0.015)	0.88
K			2.25 (1.58)
Residual variance			8970 (296.0)

The likelihood values were used to compare the Gompertz curve with two other nonlinear curve: the Logistic function and the Brody function, in a phenotypic analysis. The Brody function was defined as in equation (2) and the Logistic function was:

$$y_{ij} = \frac{A_i}{1 + B_i \exp(-K_i t_{ij})} + \epsilon_{ij} \tag{2.4.9}$$

The three nonlinear curves had the same number of parameters, and the likelihood (-2 Log L) values obtained were 696 for the Gompertz function, 1112 for the Logistic function and 3776 for the Brody function (a constant $c = 46000$ was added to the three likelihood values to make them more easily comparable). As expected, it was found that the Gompertz function was more appropriate to model this growth phenomenon. It is useful, however, to have a likelihood criterion for nonlinear model comparisons when a less well known character is analyzed. Any nonlinear function can be defined in the available SAEM program.

Phenotypic analyses of these data with the Gompertz function were also performed with winBUGS (Spiegelhalter et al., 2004), for a Bayesian Gibbs Sampling analysis. Many convergence problems were encountered, especially for fitting different line effects for the B parameter, and the algorithm showed a great sensitivity to the choice of the prior distributions. On the other hand, the SAEM algorithm proved to be more robust to the choice of starting values and showed a much faster convergence.

2.5 Discussion

The Stochastic Approximation EM (SAEM) algorithm presented in this work is conceptually very simple and has several advantages compared to a classical Monte Carlo EM algorithm (Wei and Tanner, 1990b). Firstly, thanks to the "recycling" of the simulated values from one iteration to the next, it considerably reduces the number of Monte Carlo simulations required. Secondly, the smoothing parameter considerably accelerates convergence to the MLEs. Comparison of the SAEM algorithm with approximated estimation procedures such as First Order Conditional Estimation (FOCE), Laplacian methods or the Gaussian quadrature (Davidian and Giltinan, 2003) was performed by Kuhn and Lavielle (2005). The SAEM algorithm was found to perform better than the other methods in terms of robustness with regard to the choice of the starting values, especially for the variance components, and accuracy of the estimates. It is also much faster to converge than classical Bayesian methods using the Gibbs sampling. These properties of the SAEM algorithm were confirmed here in the simulation study. The SAEM algorithm is implemented in a specialized software for the phenotypic analysis of nonlinear mixed effects models called "Monolix", which can be freely downloaded from the following address:

http://www.math.u-psud.fr/~lavielle/monolix/logiciels. A Matlab program for the sire model extension is available.

Another advantage of the stochastic EM algorithm is that it remains within the maximum likelihood framework, and therefore allows to use classical model comparison criteria such as AIC or BIC. It is possible, in particular, to compare nonlinear mixed models to other longitudinal models such as random regression or structured antedependence models. In this study, for example, it was found that the structured antedependence models (Nunez-Anton and Zimmerman, 2000; Jaffrezic et al., 2004) were better able to fit the covariance structure than the nonlinear Brody function. This shows that it might be necessary to define more flexible nonlinear functions for growth curves, which would still have interpretable parameters in terms of adult body weight and maturation rates, but would have additional parameters to capture better the variance and correlation patterns of the data. For example, functions defined by differential equations might be more appropriate. Indeed, extension of the SAEM algorithm for differential equation models is under investigation for phenotypic analyses. It was also found that, although mathematically equivalent, different parameterizations of the growth curve models (Brody, Gompertz, Richards) may improve convergence.

The aim of this work was to present this novel and efficient estimation procedure, namely the SAEM algorithm. It was applied here for the genetic analysis of nonlinear longitudinal characters such as growth traits. This algorithm is, however, very general and can also be extended for estimation in the context of mixture models, for the classification of genes with regard to their expression profile dynamics, for example. Or, it can be used for inference in generalized linear mixed models (GLMM), for the analysis of categorical traits such as fertility, or the joint analysis of discrete and continuous variables for the genetic analysis of disease resistance characters. Another extension of the SAEM algorithm could also be for QTL detection for nonlinear traits, such as growth trajectories (Ma et al., 2002), or for QTL detection of discrete traits such as disease resistance characters.

It was found that the speed of convergence of the SAEM algorithm can be improved by the use of a PX modification (Lavielle and Meza, 2006). This proved to be particularly efficient during the first iterations, when the parameters were highly correlated, as it is the case for growth curve models. A REML extension of the SAEM algorithm is proposed by Meza et al. (2006) in the phenotypic case and proved to improve the accuracy of the variance parameter estimates in similar proportions as in linear mixed models.

Chapitre 3

A Parameter Expansion version of the SAEM algorithm

Summary

The EM algorithm and its extensions are very popular tools for maximum likelihood estimation in the incomplete data setting. One of the limitations of these method is their slow convergence. The PX-EM (parameter–expanded EM) algorithm was proposed by Liu, Rubin and Wu to make EM much faster. On the other hand, stochastic versions of EM are powerful alternatives of EM when the E-step of EM is untractable in a closed form. We propose in this paper the PX-SAEM which is a parameter expansion version of the so-called SAEM (Stochastic Approximation version of EM). PX-SAEM is shown to accelerate SAEM and improve convergence toward the maximum likelihood estimate in a parametric framework. Numerical examples illustrate the behavior of PX–SAEM in linear and nonlinear mixed effects models.

This chapter corresponds, with more details, to an article which is a joint work with Marc Lavielle accepted to *Statistics and Computing*.

Contents

3.1 Introduction

The so-called incomplete-data (or partially-observed-data) models are statistical models which involve observed and unobserved data. The standard incomplete-data scheme considers the observable incomplete data $y \sim p(y;\theta)$ to result from partial observation of complete data $(y,\phi) \sim f(y,\phi;\theta)$, where p and f are some known density functions. Maximum likelihood estimation of θ consists in computing the value of θ that maximizes the observed likelihood $p(y;\theta)$. Maximum likelihood estimation for these models is a difficult challenge since the likelihood of the observations cannot usually be computed in closed form.

The Expectation-Maximization (EM) algorithm, proposed by Dempster, Laird and Rubin (1977), is a broadly applicable approach for the iterative computation of maximum likelihood estimates, useful in a variety of incomplete-data statistical problems. The E-step of the EM algorithm computes $Q(\theta|\theta_k) = \mathrm{E}[\log f(y,\phi;\theta)|y;\theta_k]$ and the M-step determines θ_{k+1} as maximizing $Q(\theta|\theta_k)$. Then, the observed-data likelihood sequence $(p(y;\theta_k))$ is non-decreasing along any EM sequence, see Wu (1983) for more details. Many extensions and

variations of EM algorithm are can be found in the literature (see McLachlan and Krishnan, 1997, and the many references therein). For example, Meng and Rubin (1993a) replace the M-step by a sequence of conditional maximizations (CM) of the complete log–likelihood. This modification of EM is called ECM (Expectation Conditional Maximization). A generalization of the ECM algorithm, called ECME, can be obtained by replacing some CM-steps of ECM (Liu and Rubin, 1994). Fesslern and Hero (1994) propose a space-alternating generalized EM (SAGE) algorithm, which updates the parameters sequentially using a sequence of small hidden data spaces, rather than simultaneously using one large complete-data space.

Stochastic versions of EM have been introduced from different perspectives to deal with situations where the E-step is infeasible in closed form. Celeux and Diebolt (1985) propose the Stochastic EM algorithm (SEM), in which the E-step is a stochastic mean on simulated missing data. Monte Carlo EM (MCEM) replaces this step by a Monte Carlo approximation based on a large number of independent simulations of the missing data, see Wei and Tanner (1990a). An extension of MCEM is proposed by Jank (2004) and is based on the Quasi–Monte Carlo methods. These methods produce deterministic sequences of points that can significantly improve the accuracy of Monte Carlo approximations over purely random sampling. Then, the Quasi–Monte Carlo EM (QMCEM) consists in the implementation of an automated, data-driven Monte Carlo EM algorithm based on randomized Quasi-Monte Carlo. Recently, Wu (2004) emphasizes that MCEM is computationally intensive.

As an alternative to reduce the amount of required simulations by MCEM and QMCEM, the SAEM algorithm, proposed by Delyon et al. (1999), replaces the E–step by a stochastic approximation. The k-th step of SAEM generates m realizations $\phi^{(k+1,l)}$ $(1 \leq l \leq m)$ from $p(\phi|y;\theta_k)$ and updates $Q_k(\theta)$ according to

$$Q_{k+1}(\theta) = Q_k(\theta) + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\log f(y,\phi^{(k+1,l)};\theta) - Q_k(\theta)\right), \tag{3.1.1}$$

where (γ_k) is a sequence of positive step sizes decreasing to 0. Then the M-step consists in determining θ_{k+1} which maximizes $Q_{k+1}(\theta)$. Precise results of convergence of SAEM are presented in Delyon et al. (1999). These results were extended by Kuhn and Lavielle (2005) when a Markov Chain Monte Carlo (MCMC) procedure is used for the simulation of the sequences $(\phi^{(k+1,l)})$.

A major drawback of EM–type procedures is their slow convergence in some situations. The parameter expansion method PX–EM was proposed by Liu et al. (1998) to

make EM faster. The idea of PX–EM is to introduce an expanded complete–data model with a larger set of parameters in the E–step and a reduction function to return to the original model in the M–step. This algorithm can be interpreted as a 'covariance adjustment' to correct the M–step, capitalizing on extra information captured in the imputed complete data. Then, PX–EM has a rate of convergence at least as fast as the standard EM because its M–step performs a more efficient analysis by fitting the expanded model.

In order to improve the convergence of the stochastic approximation version of EM, we propose to adapt the PX–EM algorithm to the SAEM algorithm. Section 3.2 describes the standard EM and PX–EM algorithms and the stochastic versions SAEM and PX–SAEM. Section 3.3 is dedicated to the application of these algorithms to linear and nonlinear mixed effects models. Some numerical experiments are proposed in Section 3.4. These examples show that the PX–SAEM algorithm improves substantially the speed of convergence toward the maximum likelihood estimate and the differences in computing time required per iteration between SAEM and PX–SAEM are negligible. Furthermore, the simulation study also emphasizes that the PX-SAEM algorithm allows to avoid local maxima of the likelihood.

3.2 The algorithms

3.2.1 The EM and PX-EM algorithms

We consider an incomplete data model where y is the set of observed data and ϕ the set of non observed data. We assume that the likelihood of the complete data (y,ϕ) depends on some parameter θ. The EM algorithm maximizes the observed likelihood $p(y;\theta)$ by iteratively maximizing the expectation of the conditional log-likelihood of the complete data: at iteration k, the E–step computes $Q_{k+1}(\theta) = \mathbb{E}(\log f(y,\phi;\theta)|y,\theta_k)$ whereas the M–step computes θ_{k+1} by maximizing $Q_{k+1}(\theta)$.

The EM sequence (θ_{k+1}) converges to a stationary point of the observed likelihood under general regularity conditions (Dempster et al., 1977; Wu, 1983).

Liu et al. (1998) proposed the PX-EM (parameter expanded EM) to accelerate the EM algorithm. The PX-EM algorithm expands the complete-data model parametrized by θ, to a larger model parametrized by Θ, with $\Theta = (\theta,\alpha)$ and where α is a kind of working parameter. Furthermore, there exists a many–to–one reduction function $R: \Theta \to R(\Theta)$ which preserves the original observed-data model, and a value α_0 of α that preserves the original complete-data model (see Liu et al., 1998 for more details).

The PX–EM algorithm uses the expanded complete-data model $f_X(y,\phi;\Theta)$ in the E-step and the reduction function in the M-step to return to the original model $f(y,\phi;\theta)$. More precisely, iteration k of PX-EM consists in the following two steps:

- *PX–Expectation step*: compute the conditional expectation of the complete log-likelihood
$$Q_{k+1}(\Theta) = \mathbb{E}(\log f_X(y,\phi;\Theta)|y;\Theta_k = (\theta_k,\alpha_0))$$
- *PX Maximisation–step*: compute $\widehat{\Theta}_{k+1}$ that maximizes $Q_{k+1}(\Theta)$ and apply the reduction function to obtain $\theta_{k+1} = R(\widehat{\Theta}_{k+1})$ and $\Theta_{k+1} = (\theta_{k+1},\alpha_0)$.

An explicit interpretation of PX-EM, as a *covariance adjustment*, is given by Liu et al. (1998). Indeed, we have:
$$\theta_{k+1}^{PX} - \theta_{k+1}^{EM} \approx b_{\theta|\alpha}(\alpha^{(k+1)} - \alpha_0) \tag{3.2.2}$$
where θ_{k+1}^{PX} is the PX–EM value at iteration $k+1$, θ_{k+1}^{EM} is the EM iterate, and $b_{\theta|\alpha}$ is a correction factor. The E-step of both algorithms consists in computing an expectation under a wrong model (using θ_{k+1} instead of θ_{ML}). Then, the M-step of PX-EM uses the difference between the imputed value of α (that can be seen as a covariate) and its true value α_0 to correct the unadjusted estimation of θ, θ_{k+1}^{EM}.

Since the monotony property of EM is preserved, Liu et al. (1998) obtained the following theorems.

Theorem 3.2.1 *PX–EM increases the loglikelihood of the observed–data model at each iteration, i.e.* $\log p(y;\theta_{k+1}) \geq \log p(y;\theta_k)$ *for all* k.

Conditions for the convergence of PX–EM iterates to a stationary point or a local maximum, θ_{MLE}, can be obtained following Wu (1983). Furthermore, Liu et al. (1998) shown also the following result:

Theorem 3.2.2 *Given that PX–EM converges to* (θ_{MLE},α_0), *and the derivatives and inverses used in the above derivations exist, PX–EM dominates EM in global rate of converge.*

3.2.2 The SAEM and PX-SAEM algorithms

Description of the algorithms

The standard SAEM algorithm, proposed by Delyon et al. (1999) consists in replacing the usual E–step of EM by a stochastic approximation procedure. The M-step is unchanged.

Following PX-EM, the PX-SAEM algorithm is a parameter expansion version of SAEM. Each iteration of PX-SAEM is decomposed into three steps: the Simulation–step and the Stochastic Approximation step of SAEM using the expanded model and the PX-M-step of PX-EM. Thus, at iteration k,

---

- *PX Simulation–step*: draw $\phi^{(k+1)}$ from the conditional distribution $p_X(\cdot|y;\Theta_k = (\theta_k,\alpha_0))$.

- *PX Stochastic approximation–step*: update $Q_{k+1}(\Theta)$ according to

$$Q_{k+1}(\Theta) = Q_k(\Theta) + \gamma_{k+1}\left(\log f_X(y,\phi^{(k+1)};\Theta) - Q_k(\Theta)\right) \tag{3.2.3}$$

  where (γ_k) is a decreasing sequence of positive numbers.

- *PX Maximisation–step*: compute $\widehat{\Theta}_{k+1}$ that maximizes $Q_{k+1}(\Theta)$ and apply the reduction function to obtain $\theta_{k+1} = R(\widehat{\Theta}_{k+1})$ and $\Theta_{k+1} = (\theta_{k+1},\alpha_0)$.

Simulation–step

When the simulation–step cannot be directly performed, Kuhn and Lavielle (2004) propose to combine this algorithm with a Markov Chain Monte Carlo (MCMC) procedure: the sequence $(\phi^{(k)})$ is a Markov Chain with transition kernels (Π_{Θ_k}) and the Simulation–step becomes:

- *Simulation–step*: using $\phi^{(k)}$, draw $\phi^{(k+1)}$ from transition probability $\Pi_{\Theta_k}(\phi^{(k)},\cdot)$.

In the practice, the Hasting–Metropolis algorithm is used to obtain an approximation of these conditional distributions. Furthermore, we ran m Markov Chains to improve the convergence of the algorithm. Then, the simulation step requires to draw m sequences

$\phi^{(k+1,1)}, \ldots, \phi^{(k+1,m)}$ at the iteration k and to combine stochastic approximation and Monte Carlo in the approximation step:

$$s_{k+1} = s_k + \gamma_{k+1} \left(\frac{1}{m} \sum_{l=1}^{m} \tilde{S}(y,\phi^{(k+1,l)}) - s_k \right). \tag{3.2.4}$$

Convergence of PX–SAEM

The PX version of SAEM is useful only during the first iterations of the algorithm, in order to converge in few iterations to a neighborhood of the maximum of the observed likelihood. Following Kuhn and Lavielle (2005), in practice, it is useful to choose the first stepsizes equal to 1, in order to allow more flexibility during the first iteration. In general the initial guess θ_0 may be far from the maximum likelihood value we are looking for and the first iterations will require big variations of the sequences (θ_k). After converging to a neighborhood of the MLE, it is interesting to choose smaller stepsizes in order to refine the estimation near the objective value. The almost sure convergence of the algorithm to the MLE is then ensured by using a decreasing sequence (γ_k) without any parameter expansion. In practice, we recommend to set $\gamma_k = 1$ for $1 \leq k \leq K$ and $\gamma_k = (k-K)^{-1}$ for $K \geq k+1$.

It was shown by Delyon et al. (1999) and by Kuhn and Lavielle (2004) that SAEM converges to a maximum (local or global) of the likelihood of the observations under very general conditions. Convergence of PX-SAEM is ensured under the same conditions since the parameter expansion is introduced only during the first iterations of the algorithm. To choose the number of iterations where the expanded model is used, we must consider that the M step under the expanded model can be more complex and this situation may work against the speed increase produced by the PX-SAEM algorithm. In practice, few iterations of PX-SAEM - about 10 - are enough to obtain a very fast convergence.

3.2.3 Application of these algorithms to exponential models

We will assume here that the expanded complete–data model belongs to the exponential family. Then, the expanded complete log-likelihood has the form:

$$\log f_X(y,\phi;\Theta) = -\Psi(\Theta) + \langle S(y,\phi),\xi(\Theta) \rangle$$

where $S(y,\phi)$ is the sufficient statistics of the expanded complete–data model.

The E-step of EM reduces to computing

$$s_{k+1} = \mathbb{E}\left(S(y,\phi)|y;\Theta_k\right) \tag{3.2.5}$$

and the SA-step of SAEM reduces to computing

$$s_{k+1} = s_k + \gamma_{k+1}\left(S(y,\phi^{(k+1)}) - s_k\right) \tag{3.2.6}$$

The PX–Maximization steps of PX-EM and PX-SAEM are identical: compute

$$\widehat{\Theta}_{k+1} = \operatorname{Arg}\max_{\Theta}\{-\Psi(\Theta) + \langle s_{k+1},\xi(\Theta)\rangle\}$$

and apply the reduction function to obtain $\theta_{k+1} = R(\widehat{\Theta}_{k+1})$ and $\Theta_{k+1} = (\theta_{k+1},\alpha_0)$.

3.3 Application to the mixed effects model

3.3.1 Linear mixed effects model

We consider the following linear model:

$$\boldsymbol{y_i} = \boldsymbol{X_i}\boldsymbol{\beta} + \boldsymbol{Z_i}\boldsymbol{\phi_i} + \boldsymbol{\varepsilon_i} \quad 1 \leq i \leq N, \tag{3.3.7}$$

where $\boldsymbol{y_i}$ is a $n_i \times 1$ observed vector, $\boldsymbol{\beta}$ is a $p \times 1$ unknown vector, $\boldsymbol{\phi_i} \sim \mathcal{N}(0,\Gamma)$ is a $d \times 1$ non observed random vector, $\boldsymbol{X_i}$ and $\boldsymbol{Z_i}$ are two $n_i \times p$ and $n_i \times d$ known matrix and $\boldsymbol{\varepsilon_i} \sim \mathcal{N}(0,\sigma^2 Id_{n_i})$. Furthermore, we suppose that $(\boldsymbol{\varepsilon_i})$ and $(\boldsymbol{\phi_i})$ are mutually independent. Our goal is to compute the maximum likelihood estimator of the unknown parameters vector $\boldsymbol{\theta} = (\boldsymbol{\beta},\Gamma,\sigma^2)$.

Let $\boldsymbol{y} = (\boldsymbol{y_1},\boldsymbol{y_2},\ldots,\boldsymbol{y_N})$, $\boldsymbol{\phi} = (\boldsymbol{\phi_1},\boldsymbol{\phi_2},\ldots,\boldsymbol{\phi_N})$ and $N_{tot} = \sum_{i=1}^{N} n_i$ be the total number of observations. Then, the complete likelihood can be written as

$$\begin{aligned}\log f(\boldsymbol{y},\boldsymbol{\phi};\boldsymbol{\theta}) &= -\frac{N}{2}\log(2\pi|\Gamma|) - \frac{N_{tot}}{2}\log(2\pi\sigma^2) - \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{\phi_i}'\Gamma^{-1}\boldsymbol{\phi_i} \\ &\quad -\frac{1}{2\sigma^2}\sum_{i=1}^{N}||\boldsymbol{y_i} - \boldsymbol{X_i}\boldsymbol{\beta} - \boldsymbol{Z_i}\boldsymbol{\phi_i}||^2,\end{aligned}$$

and we deduce that:

$$\boldsymbol{\phi_i}|\boldsymbol{y_i};\boldsymbol{\theta} = \sim \mathcal{N}(\boldsymbol{m_i},\boldsymbol{V_i})$$

where

$$V_i = \left(\frac{Z_i'Z_i}{\sigma^2} + \Gamma^{-1}\right)^{-1} \quad ; \quad m_i = V_i \frac{Z_i'(y_i - X_i\beta)}{\sigma^2}.$$

Thus, the Expectation–step of EM and the Simulation–step of SAEM can be easily performed:

- *E-step* of EM:

$$\begin{aligned} s_{1,i,k+1} &= E(\phi_i|y_i;\theta_k), \\ s_{2,i,k+1} &= E(\phi_i\phi_i'|y_i;\theta_k) \\ &= s_{1,i,k+1}s_{1,i,k+1}' + \mathrm{Cov}\,(\phi_i|y_i;\theta_k)) \end{aligned}$$

- *SA–step* of SAEM:

$$\begin{aligned} s_{1,i,k+1} &= s_{1,i,k} + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\phi_i^{(k+1,l)} - s_{1,i,k}\right) \\ s_{2,i,k+1} &= s_{2,i,k} + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\phi_i^{(k+1,l)}\phi_i^{(k+1,l)'} - s_{2,i,k}\right) \end{aligned}$$

The Maximization–step is the same for both algorithms:

$$\begin{aligned} \beta_{k+1} &= \left(\sum_{i=1}^{N} X_i'X_i\right)^{-1}\sum_{i=1}^{N} X_i'(y_i - Z_i s_{1,i,k+1}) \\ \Gamma_{k+1} &= \frac{1}{N}\sum_{i=1}^{N} s_{2,i,k+1} \\ \sigma^2_{k+1} &= \frac{1}{N_{tot}}\sum_{i=1}^{N} ||y_i - X_i\beta_{k+1}||^2 + \mathrm{tr}\,(Z_i s_{2,i,k+1} Z_i') - 2\,(y_i - X_i\beta_{k+1})'\,Z_i s_{1,i,k+1} \end{aligned}$$

where $\mathrm{tr}\,(A)$ is the trace of matrix A.

To perform the PX algorithms in mixed models context, we must specify how to introduce the working parameter α and define the reduction function R as well as the value α_0 of α which preserves the original model. The comments concerning PX-EM in Liu et al. (1998) also hold for PX-SAEM: the updated parameter Γ_{k+1} depends on the imputed matrix $(\phi_i^{(k+1,\ell)}\phi_i^{(k+1,\ell)\prime})$ and the model covariance between y_i and ϕ_i is fixed at $Z_i\Gamma_{k+1}$. To adjust for the deviations between this covariance and the relationship between the imputed random effects $\phi_i^{(k+1,\ell)}$ and the known y_i, we introduce the working parameter

as a $d \times 1$ vector $\alpha = (\alpha_1, \cdots, \alpha_d)$ which we incorporate into the model by reshaping the random effects. Then, the expanded model for the PX–(SA)EM algorithms, is:

$$y_i = X_i\beta + Z_i D_\alpha \phi_i + \varepsilon_i, \quad 1 \leq i \leq N,$$

where D_α is the diagonal $d \times d$ matrix formed with vector α. Denoting D_{ϕ_i} the $d \times d$ diagonal matrix formed with vector ϕ_i, we can write the expanded model as follows:

$$y_i = (X_i \quad Z_i D_{\phi_i}) \begin{pmatrix} \beta \\ \alpha \end{pmatrix} + \varepsilon_i.$$

Thus, $\Theta = (\beta,\Gamma,\sigma^2,\alpha)$ and the reduction function is $R(\Theta) = (\beta,D_\alpha \Gamma D_\alpha,\sigma^2)$. The original model is obtained with $\alpha_0 = (1,1,\ldots,1)$, that is with $D_{\alpha_0} = Id_d$.

In both algorithms, only the Maximization–step change. At iteration $k+1$, we compute $\widehat{\Theta}_{k+1} = (\beta_{k+1},\Gamma_{k+1},\sigma^2_{k+1},\alpha_{k+1})$ as follows:

Let

$$\begin{aligned} A_i &= \begin{pmatrix} X_i'X_i & X_i'Z_i D_{s_{1,i,k+1}} \\ D_{s_{1,i,k+1}} Z_i' X_i & Z_i' Z_i s_{2,i,k+1} \end{pmatrix} \\ B_i &= \begin{pmatrix} X_i & Z_i D_{s_{1,i,k+1}} \end{pmatrix}. \end{aligned}$$

Thus,

$$\begin{aligned} \begin{pmatrix} \beta_{k+1} \\ \alpha_{k+1} \end{pmatrix} &= \left(\sum_{i=1}^N A_i \right)^{-1} \sum_{i=1}^N B_i' y_i \\ \Gamma_{k+1} &= \frac{1}{N} \sum_{i=1}^N s_{2,i,k+1} \\ \sigma^2_{k+1} &= \frac{1}{N_{tot}} \sum_{i=1}^N ||y_i - X_i\beta_{k+1}||^2 + \mathrm{tr}\left(Z_i D_{\alpha_{k+1}} s_{2,i,k+1} D_{\alpha_{k+1}} Z_i' \right) \\ &\quad -2\alpha_{k+1} \left(y_i - X_i\beta_{k+1} \right)' Z_i s_{1,i,k+1}. \end{aligned}$$

Then we apply the reduction function R to obtain $\theta_{k+1} = R(\widehat{\Theta}_{k+1})$ and $\Theta_{k+1} = (\theta_{k+1},(1,1,\ldots,1))$.

3.3.2 Nonlinear mixed effects model

Consider now the following mixed effects:

$$y_{ij} = g(x_{ij},\phi_i) + h(x_{ij},\phi_i)\varepsilon_{ij}, \quad 1 \leq i \leq N, \ 1 \leq j \leq n_i. \tag{3.3.8}$$

Here, (x_{ij}) is the known set of design variables and the random vector of individual parameters $\phi_i \in \mathbb{R}^d$ is modelized by:

$$\phi_i = \mu + \eta_i \quad \text{with} \quad \eta_i \sim_{i.i.d.} \mathcal{N}(0,\Gamma). \tag{3.3.9}$$

We suppose that the (ε_{ij}) and the (η_i) are mutually independent.

In this context, the set of parameters is $\theta = (\mu,\Gamma,\sigma^2)$ and the complete log–likelihood can be written as

$$\begin{aligned}\log f(y,\phi;\theta) &= -\frac{N}{2}\log(2\pi|\Gamma|) - \frac{N_{tot}}{2}\log(2\pi\sigma^2) - \frac{1}{2}\sum_{i=1}^{N}(\phi_i-\mu)'\Gamma^{-1}(\phi_i-\mu) \\ &\quad - \sum_{i=1}^{N}\sum_{j=1}^{n_i}\log(g(x_{ij},\phi_i)) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{n_i}\left(\frac{y_{ij}-g(x_{ij},\phi_i)}{h(x_{ij},\phi_i)}\right)^2.\end{aligned}$$

The model is said to be nonlinear if g is a nonlinear function of the individual random parameters ϕ_i and/or $h \neq 1$. In this case, the conditional distribution of ϕ cannot be computed in a closed form and the E-step of EM is untractable.

On the other hand, the non–observed data ϕ can be simulated with the Hasting-Metropolis algorithm. Several transition kernels, associated to different proposals can be successively used. We use the two following proposal kernels to approximate $\phi^{(k+1)}|y;\theta_k$:

1. the proposal $q^{(1)}_{\theta_k}$ is the prior distribution of ϕ_i,
2. the proposal $q^{(2)}_{\theta_k}$ is the random walk $\mathcal{N}(\phi_i^{(k)},\rho_1^2\Gamma_k)$, where ρ_1 is a constant.

Then, the Simulation–step at iteration k consists, in a first time, in running h_1 iterations of the Hasting–Metropolis algorithm with proposal $q^{(1)}_{\theta_k}$ and h_2 iterations with proposal $q^{(2)}_{\theta_k}$. The approximation step of SAEM reduces to updating the sufficient statistics of the complete model:

$$
\begin{aligned}
s_{1,k+1} &= s_{1,k} + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\sum_{i=1}^{N}\phi_i^{(k+1,l)} - s_{1,k}\right)\\
s_{2,k+1} &= s_{2,k} + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\sum_{i=1}^{N}\phi_i^{(k+1,l)}\phi_i^{(k+1,l)'} - s_{2,k}\right)\\
s_{3,k+1} &= s_{3,k} + \gamma_{k+1}\left(\frac{1}{m}\sum_{l=1}^{m}\sum_{i=1}^{N}\sum_{j=1}^{n_i}\left(\frac{y_{ij} - g(x_{ij},\phi_i^{(k+1,l)})}{h(x_{ij},\phi_i^{(k+1,l)})}\right)^2 - s_{3,k}\right).
\end{aligned}
$$

Then, θ_{k+1} is obtained in the maximization step as follows:

$$\mu_{k+1} = \frac{s_{1,k+1}}{N} \tag{3.3.10}$$

$$\Gamma_{k+1} = \frac{s_{2,k+1}}{N} - \mu_{k+1}\mu_{k+1}' \tag{3.3.11}$$

$$\sigma^2_{k+1} = \frac{s_{3,k+1}}{N_{tot}}. \tag{3.3.12}$$

Following what was proposed for the linear model, the expanded model used for the PX–SAEM algorithms is:

$$y_{ij} = g(x_{ij},\boldsymbol{D}_\alpha\phi_i) + h(x_{ij},\phi_i)\varepsilon_{ij}, \quad 1 \le i \le N, \ 1 \le j \le n_i$$

where $\boldsymbol{D}_\alpha$ is a diagonal matrix. Thus, $\Theta = (\mu,\Gamma,\sigma^2,\alpha)$, the reduction function is $R(\Theta) = (\boldsymbol{D}_\alpha\mu,\boldsymbol{D}_\alpha\Gamma\boldsymbol{D}_\alpha,\sigma^2)$ and $\alpha_0 = (1,1,\cdots,1)$.

Since the PX version of SAEM is used when $\gamma_k = 1$, there is no stochastic approximation–step during these first iterations. It can be noticed from equation (3.2.3) that $\widehat{\Theta}_{k+1}$ maximizes the complete–data log–likelihood $\log p_X(y,\phi^{(k+1)};\Theta)$ and the maximization step of PX–SAEM just reduces to

$$\alpha_{k+1} = \operatorname*{Arg\,min}_{\alpha}\sum_{i=1}^{N}\sum_{j=1}^{n_i}\left(\frac{y_{ij} - g(x_{ij},\boldsymbol{D}_{\alpha_k}\phi_i^{(k+1)})}{h(x_{ij},\boldsymbol{D}_{\alpha_k}\phi_i(k+1))}\right)^2 \tag{3.3.13}$$

$$\mu_{k+1} = \frac{1}{N}\sum_{i=1}^{N}\phi_i^{(k+1)} \tag{3.3.14}$$

$$\Gamma_{k+1} = \frac{1}{N}\sum_{i=1}^{N}(\phi_i^{(k+1)} - \mu_{(k+1)})(\phi_i^{(k+1)} - \mu_{(k+1)})' \tag{3.3.15}$$

$$\sigma^2_{k+1} = \frac{1}{N_{tot}}\sum_{i=1}^{N}\sum_{j=1}^{n_i}\left(\frac{y_{ij} - g(x_{ij},\boldsymbol{D}_{\alpha_{k+1}}\phi_i^{(k+1)})}{h(x_{ij},\boldsymbol{D}_{\alpha_{k+1}}\phi_i^{(k+1)})}\right)^2 \tag{3.3.16}$$

and to apply the reduction function R.

Remark: When f is nonlinear, α_{k+1} cannot be computed in a closed form but a minimization procedure, such as the Newton–Raphson can be used.

3.4 Numerical examples

In this section, we illustrate the PX–SAEM with two examples, a linear and a nonlinear mixed–effects models. Due to the stochastic structure of the SAEM algorithm, we do not "measure" the convergence speed increase with a convergence criterion but instead with a graphical analysis. We analyze the parameters estimation evolution over iterations but also the exact or estimated (in the nonlinear case) log–likelihoodd of the observation evolution over iterations. These graphs clearly show the converge speed improvement obtained with PX–SAEM.

3.4.1 A linear model

We consider the model define in (3.3.7) with $\boldsymbol{X_i} = (1,i)$ and $Z_i = i/10$. Here, $n_i = 1$, $p = 2$ and $d = 1$. $N = 100$ observations were simulated with $\boldsymbol{\beta} = (0,0)$, $\Gamma = \gamma_1 = 0.5$ and $\sigma^2 = 25$.

Since the model is linear, we can compare SAEM and PX–SAEM with EM and PX–EM. We ran these four algorithms with the same initial guess: $\boldsymbol{\beta_0} = (6,0.05)'$, $\gamma_0 = 20$ and $\sigma_0^2 = 50$.

As mentioned before, PX-SAEM uses the expanded model only during the first 10 iterations to improve the speed of convergence of the algorithm. Then, the standard SAEM algorithm is used with the original model after these 10 iterations. This strategy ensures the almost sure convergence of the algorithm to a maximum of the observed likelihood. On the other hand, the expanded model is used during all the iterations of the PX–EM algorithm.

The sequence of stepsizes (γ_k) used with SAEM and PX-SAEM is $\gamma_k = 1$ for $1 \leq k \leq 300$ and $\gamma_k = 1/(k-300)$ for $301 \leq k \leq 1000$. At each iteration, the number of simulated sequences for the stochastic approximation is $m = 50$.

Figures 3.1 and 3.2 show the parameter estimations (θ_k). Figure 3.3 displays the observed log–likelihoods $(\log p(y;\theta_k))$ obtained with these four algorithms along 1000 iterations.

We see that the PX versions of EM and SAEM converge much more faster than the standard EM and SAEM algorithms. This is clearly obvious for the variance components. Furthermore, Figure 3.3 shows that the maximum of the observed log–likelihood is reached in very few iterations with the PX algorithms.

We can also note in this example that SAEM (respectively PX–SAEM) behaves like EM (respectively PX–EM). The trajectory of SAEM (respectively PX–SAEM) randomly fluctuates around the trajectory of EM (respectively PX–EM) during the first iterations with $\gamma_k = 1$ and converges almost surely when γ_k decreases.

We compared the computational times using a Matlab version of the algorithms on a computer with a Pentium M processor at 2.26GHz. There is little difference between EM and PX-EM since these algorithms require respectively about 0.15 ms and 0.18 ms per iteration. There is also very little difference between the stochastic versions of these algorithms since SAEM and PX-SAEM both require about 1.5 ms per iteration, with $m = 50$ chains (and 0.4 ms with 5 chains).

3.4.2 A nonlinear pharmacokinetic model

We will consider here the model used by Kuhn and Lavielle (2005) to compare the SAEM algorithm with the SPML algorithm of Concordet and Nunez (2002). It is a kinetic population homoscedastic model used for example for analyzing the concentration obtained after a constant drug diffusion. The data were simulated according to:

$$y_{ij} = \phi_{i1}(1 - \exp[-\phi_{i2}t_j]) + \varepsilon_{ij}$$

where y_{ij} denotes the concentration on time $t_j = j$, $1 \leq j \leq n_i = M = 7$, for individual i, $1 \leq i \leq N = 50$. We assume that $\boldsymbol{\phi_i} \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Gamma})$ where $\boldsymbol{\mu} = (50,0.5)'$ and

$$\boldsymbol{\Gamma} = \begin{pmatrix} \gamma_{11} & 0 \\ 0 & \gamma_{22} \end{pmatrix} = \begin{pmatrix} 25 & 0 \\ 0 & 0.05 \end{pmatrix}.$$

Errors terms (ε_{ij}) are independent and follow a Gaussian distribution with mean zero and variance $\sigma^2 = 16$.

The expanded model used for PX–SAEM is

$$y_{ij} = \alpha_1\phi_{i1}(1 - \exp[-\alpha_2\phi_{i2}t_j]) + \varepsilon_{ij},$$

and the expanded parameter is $\Theta = (\mu_1,\mu_2,\gamma_{11},\gamma_{22},\sigma^2,\alpha_1,\alpha_2)$, with $\alpha_{10} = \alpha_{20} = 1$.

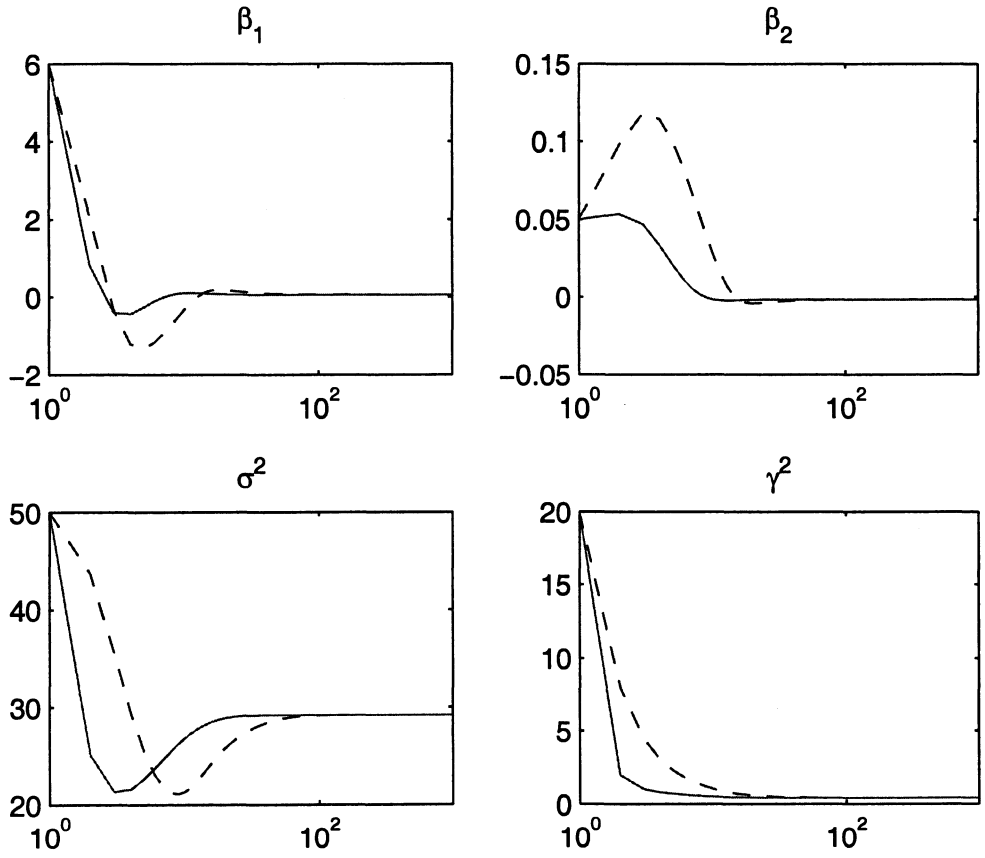


FIG. 3.1 – *The sequences* (θ_k) *using EM and PX-EM. A logarithmic scale is used for the x–axis. The PX-EM sequence is in solid line and the EM sequence in dotted line.*

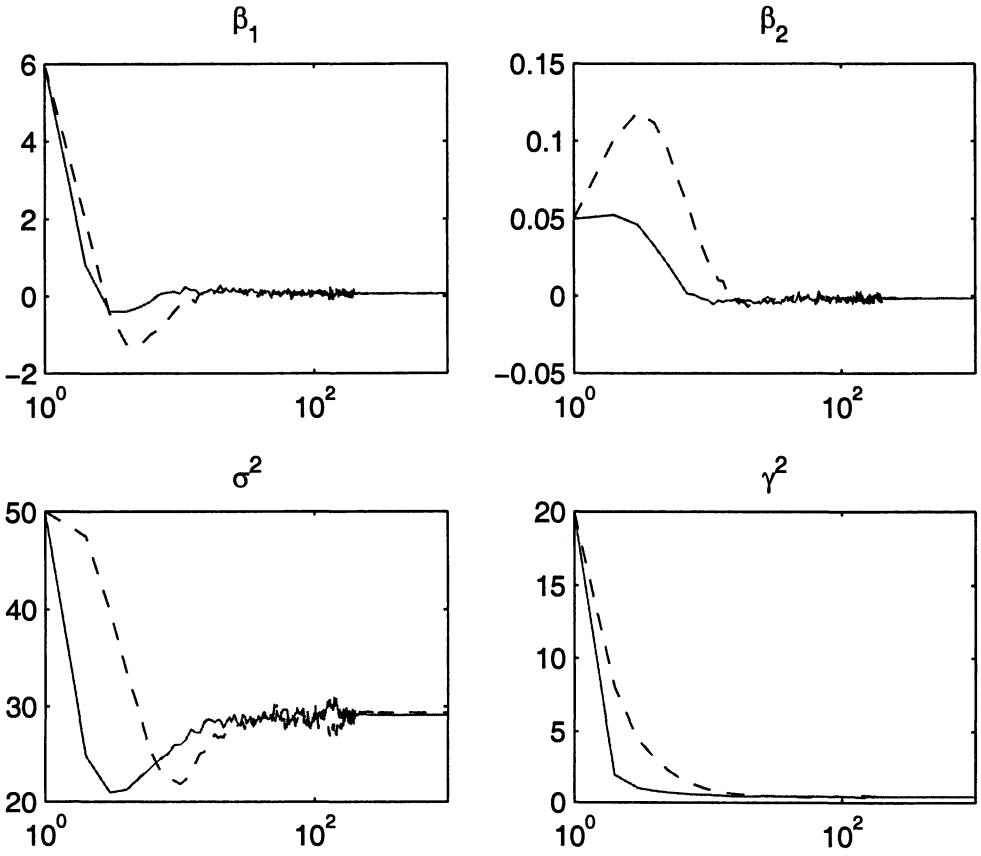


FIG. 3.2 – *The sequences* (θ_k) *using SAEM and PX-SAEM. A logarithmic scale is used for the x–axis. The PX-SAEM sequence is in solid line and the SAEM sequence in dotted line.*

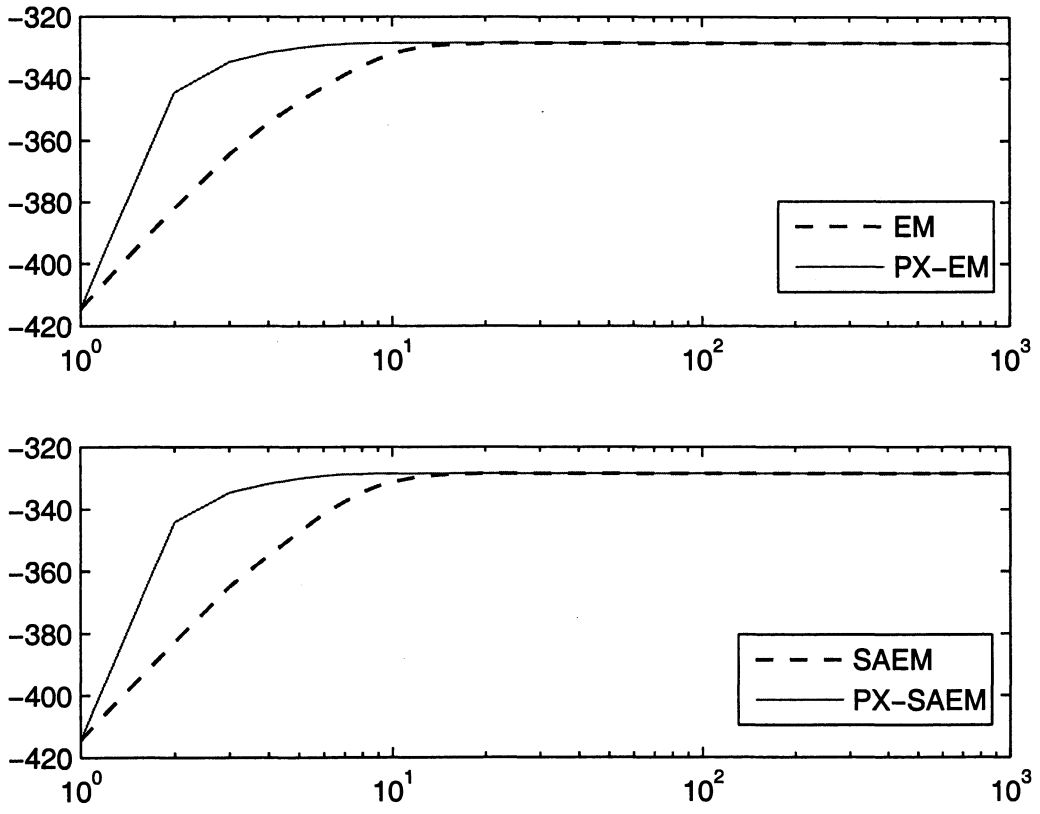


FIG. 3.3 – *The observed log–likelihood sequences* $(\log p(y;\theta_k))$ *obtained with EM and PX-EM (top), SAEM and PX-SAEM (bottom). A logarithmic scale is used for the x–axis. The PX-EM and PX-SAEM sequences are in solid line and the EM and SAEM sequences in dotted line.*

The initial values of the parameters for SAEM and PX–SAEM are

$$\mu_0 = (10,2)', \quad \Gamma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \sigma_0^2 = 60. \tag{3.4.17}$$

For both algorithms, we used 700 iterations and the following sequence (γ_k): $\gamma_k = 1$ for $1 \leq k \leq 400$ and $\gamma_k = 1/(k-400)$ for $401 \leq k \leq 700$. The number of Markov Chains used for the stochastic approximation is $m = 5$.

Here, PX–SAEM consists of the 10 iterations with the expanded model, followed with 690 iterations of the standard SAEM algorithm.

In order to compare SAEM and PX–SAEM, we simulated 100 data sets and ran both algorithms on each data set. This set of 200 runs is composed of three subsets:

1. the 80 runs of SAEM which converged to the global maximum of the observed likelihood,

2. the 20 runs of SAEM which did not converged to the global maximum of the observed likelihood,

3. the 100 runs of PX–SAEM (each one converged to the global maximum of the observed likelihood).

Figures 3.4 displays the three averaged sequences $(\boldsymbol{\theta}_k)$ computed on these three subsets of runs (the logarithms of the estimated variances are displayed instead of the variances). We clearly see in this figure that PX–SAEM converges much faster to the MLE than SAEM and also the ability of this algorithm to avoid local maxima of the likelihood.

An other way to show the performance of the PX–SAEM is study the evolution of the relative root square mean error (RMSE) over iterations. For both algorithms, we measure the relative distance between the estimation of $\boldsymbol{\theta}_k$ at iteration k and the true value $\boldsymbol{\theta}^\star$ of $\boldsymbol{\theta}$, used for the simulations. Let $(\boldsymbol{\theta}_{k,s}{}^{PX-SAEM}, 0 \leq k \leq 700)$ be the PX–SAEM sequence of estimates of $\boldsymbol{\theta}$, obtained with the s^{th} data set. Then, for each component of $\boldsymbol{\theta}$, this sequence of relative RMSE (in %) is computed as follows for PX–SAEM:

$$RMSE_k^{PX-SAEM} = 100 \times \sqrt{\frac{1}{100}\sum_{s=1}^{100}\left(1 - \frac{\boldsymbol{\theta}_{k,s}{}^{PX-SAEM}}{\boldsymbol{\theta}^\star}\right)^2}.$$

For each parameter, the three relative RMSE sequences, computed with these three different subsets of runs are displayed in Figure 3.5: we see that PX–SAEM only requires

about 10 iterations to reach the neighborhood of the MLE, while SAEM needs about 300 iterations (when it converges to the MLE). The estimated log–likelihoods are displayed in Figure 3.6. This figure also clearly shows that the maximum of the likelihood is almost reached with 10 iterations of the PX–SAEM, instead of 300 iterations with SAEM.

In the nonlinear mixed effects model case, the CPU time required for one iteration of SAEM and PX–SAEM are slightly different, because of the maximization step. We compared these computing times using a Matlab version of the algorithms on a computer with a Pentium M processor at 2.26GHz. With only $m = 5$ chains, SAEM spends 6 ms per iteration and PX–SAEM spends 12 ms. Thus, a good estimation of the parameters for this example is obtained after 120 ms (10 iterations) with PX-SAEM and after 1800 ms (300 iterations) with SAEM.

3.5 Conclusion

This work shows that the PX–SAEM can be implemented in mixed–effects model context whith positif results. We proposed to use a hybrid algorithm which applies PX–SAEM in the first iterations and then the standard SAEM algorithm. This strategy permits to ensure the convergence of the algorithm to the maximum likelihood estimates. Our graphical study demonstrates that the PX–SAEM permits to improve clearly the convergence speed of SAEM in nonlinear mixed models but also permits to avoid local maxima of the likelihood.

The PX–M step in nonlinear case is more complex and it is necessary to use maximization methods. In terms of CPU time per iteration, the differences between SAEM and PX–SAEM are negligible. Our study confirms that few iterations of PX–SAEM are necessary to obtain a speed convergence increase.

A possible extension is to combine the PX–SAEM with the Restricted Maximum Likelihood (REML) method to estimate the variance components in nonlinear mixed–effects models. Meza et al. (2006) introduced the REML estimation of variance parameters using the SAEM algorithm in these models with a significant reduction of the biais. In the other hand, Foulley and van Dyk (2000) used the PX–EM to compute REML estimates of variance components in linear mixed effects models confirming the potential advantage of this algorithm. It appear interesting to propose a REML version of PX–SAEM algorithm, which would allow to speed up the classic SAEM and to obtain unbiased variance components estimates in some situations.

FIG. 3.4 – *Estimation of θ using SAEM and PX-SAEM/SAEM. A logarithmic scale is used for the x-axis. The average of the 100 PX-SAEM runs are in solid line; the average of the 80 SAEM runs which converged to the MLE are in dotted line; the average of the 20 SAEM runs which did not converge to the MLE are in dashed line.*

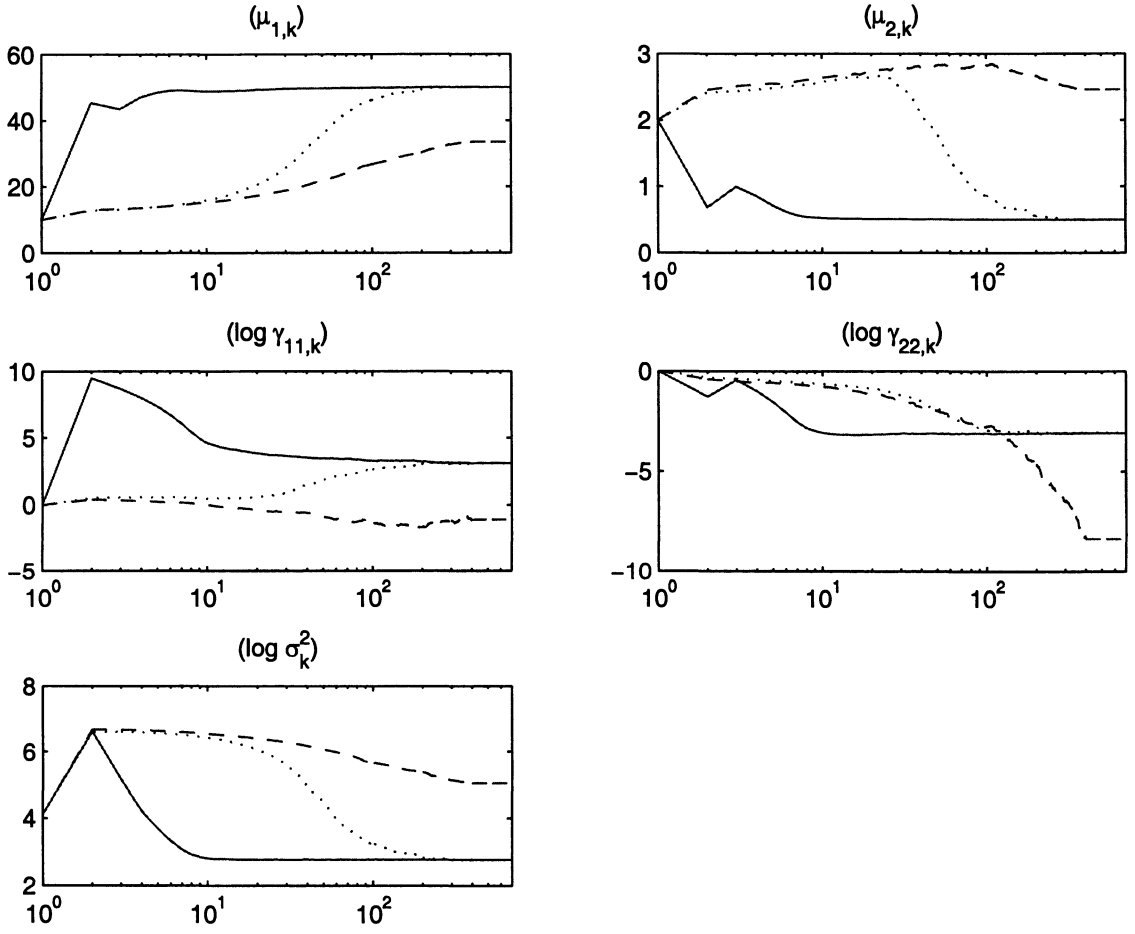


FIG. 3.5 – *Sequence of the RMSE using SAEM and PX-SAEM/SAEM. A logarithmic scale is used for the x-axis. The average of the 100 PX-SAEM runs are in solid line; the average of the 80 SAEM runs which converged to the MLE are in dotted line; the average of the 20 SAEM runs which did not converge to the MLE are in dashed line.*

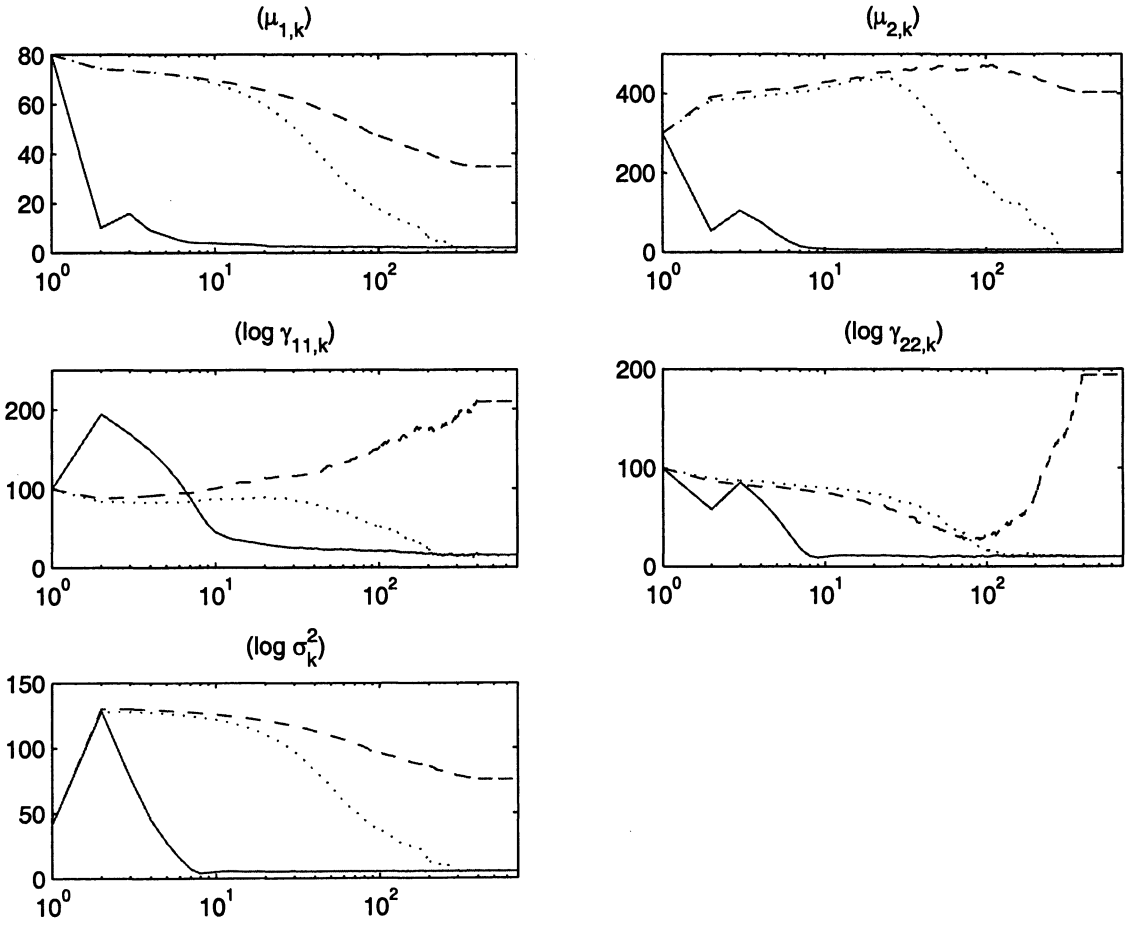
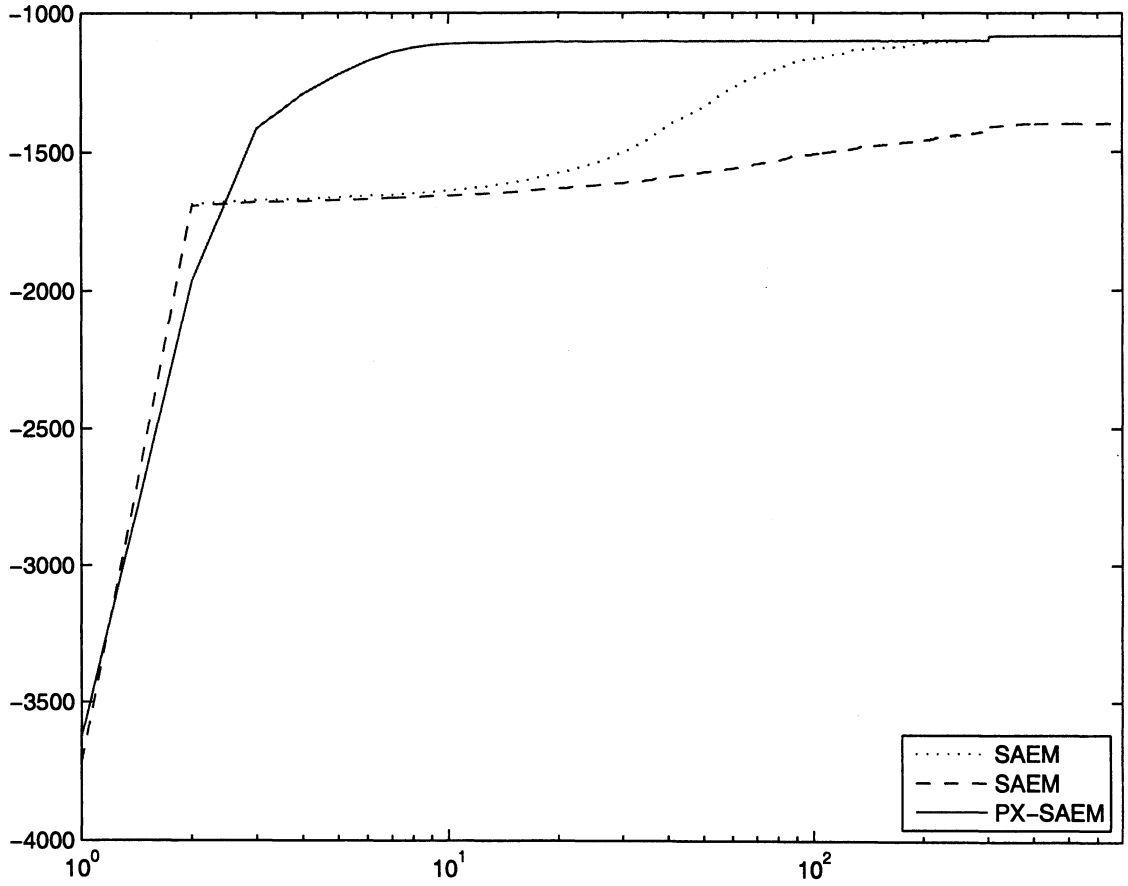


FIG. 3.6 – *The estimated observed log–likelihood sequences* ($p(\boldsymbol{y}|\boldsymbol{\theta}_k)$). *The average of the 100 PX–SAEM runs are in solid line; the average of the 80 SAEM runs which converged to the MLE are in dotted line; the average of the 20 SAEM runs which did not converge to the MLE are in dashed line.*



Chapitre 4

REML estimation of variance parameters in nonlinear mixed effects models using the SAEM algorithm

Summary

Nonlinear mixed effects models are now widely used in biometrical studies, especially in pharmacokinetic research or for the analysis of growth traits for agricultural and laboratory species. Most of these studies, however, are often based on ML estimation procedures, which are known to be biased downwards, especially for the variance components. A few REML extensions have been proposed, mainly for approximated methods. The aim of this work is to propose a novel REML estimation procedure for these models, based on an integration of the fixed effects and a stochastic estimation procedure. This method was implemented via the SAEM algorithm (Stochastic Approximation EM algorithm), which proved to be much faster than the classical Monte Carlo EM algorithm thanks to a recycling of the simulated variates from one iteration to the next. A simulation study showed that the proposed REML estimation procedure considerably reduced the bias observed with the ML estimation, as well as the residual mean squared error of the variance parameter estimations, especially in the unbalanced cases.

This chapter corresponds, with more details, to an article which is a joint work with Florence Jaffrézic and Jean-Louis Foulley submitted to *Biometrical Journal*

4.1 Introduction

Analysis of longitudinal data is an essential issue in biometrical studies, and many methodologies have already been proposed in the linear mixed model framework to analyze such data (Diggle et al., 1994; Verbeke and Molenberghs, 2000).

Focus is in this work on nonlinear mixed effects models. With the development of novel estimation procedures (Davidian and Giltinan, 2003), they are now widely used in longitudinal studies. Their main field of application is in pharmacokinetic research, to analyze within-subject pharmacokinetic processes of absorption, distribution and elimination governing the drug concentrations. They have also been widely applied for the modelling of growth traits for various agricultural and laboratory species such as mice, chicken, cattle, pigs and trees.

Classical estimation procedures developed in the context of linear mixed effects models cannot be applied in the nonlinear framework. At present, the most popular estimation procedures, implemented in standard softwares, are approximated methods based on a linearization of the likelihood via Taylor series expansion (Lindstrom and Bates, 1990). The main disadvantage of these approximated methods is that they can produce inconsistent estimates, in particular when the number of measurements per subject is not large enough.

Due to the complex form of the likelihood function, proper inference for nonlinear mixed effects models require the use of stochastic procedures. The most commonly used are Bayesian methods, implemented via for example the Gibbs Sampling. These methods are, however, extremely time consuming and are quite sensitive to the choice of prior distributions.

On the other hand, McCulloch (1997) proposed using a hybrid algorithm combining a Markov Chain Monte Carlo EM algorithm -MCEM - (Wei and Tanner, 1990b), which converges rapidly to the neighbourhood of the MLE but then shows a great deal of variability within this neighbourhood, and a Markov Chain Monte Carlo (MCMC) integration and maximization of the likelihood to obtain accurate estimates and confidence intervals. This last approach is, however, very computationally expensive and requires a reference point close to the actual MLE (Pletcher and Jaffrézic, 2002).

Recently, Kuhn and Lavielle (2004), proposed a new version of the stochastic EM algorithm, namely the Stochastic Approximation EM (SAEM) which combines the advantages of both methodologies presented above: it is quite robust to the choice of starting values, converges quite rapidly to the MLE and allows the calculation of the likelihood value and

standard errors of the parameters. These authors have, however, only developed an ML version of this algorithm.

Most of the studies using nonlinear mixed effects models are often based on ML estimation. It is well known, however, that the maximum likelihood estimator of variance components can be biased downwards because it does not adjust for the degrees of freedom lost by estimating the fixed effects. Restricted maximum likelihood (REML) corrects this problem by maximizing the likelihood of a set of residual contrasts. But Patterson and Thompson's original formulation of REML (Patterson and Thompson, 1971) does not directly extend beyond linear models, as zero-mean residual contrasts generally do not exist in nonlinear models. There are two ways to overcome this difficulty. A first possibility, as proposed by Liao and Lipsitz (2002) in the context of generalized linear mixed models, is to correct the bias in the profile score function of the variance components. This algorithm, however, proved to be extremely time consuming. It requires to integrate out the random effects, use simulation to estimate the bias and then adjust for the bias. A second possibility to obtain REML estimates is to integrate out the fixed effects (Harville, 1974). This approach will be used here. There are two ways, in theory, to perform this integration: first using the Gaussian quadrature and second via stochastic methods. In the context of nonlinear mixed effects models, however, integration via the Gaussian quadrature is extremely difficult due to the increase of dimensionality.

The aim of this work is to present a natural REML implementation for estimation in nonlinear mixed effects models within an exact estimation scheme, using integration of the fixed effects via a stochastic EM algorithm (Delyon et al., 1999). A simulation study is presented to illustrate the properties of the REML estimates compared to ML in nonlinear mixed effects models. The proposed estimation procedure is applied to the modelling of growth curves in chicken, in a genetic study.

4.2 Methodology

4.2.1 Presentation of the model

We consider the following general mixed–effects model:

$$y_{ij} = g(\phi_i, t_{ij}) + \varepsilon_{ij}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq n_i, \tag{4.2.1}$$

where $y_{ij} \in \mathbb{R}$ denotes the jth observation of subject i, t_{ij} are known design variables ($t_{ij} \in \mathbb{R}^p$), usually time, N is the number of subjects and n_i is the number of observations

of subject i. The within–group errors (ε_{ij}) are supposed to be *i.i.d.* Gaussian random variables with mean zero and unknown variance σ^2.

The model will be nonlinear when g is a nonlinear function of the individual random parameters ϕ_i. Vector $\phi_i \in \mathbb{R}^d$ is modelled by:

$$\phi_i = X_i\beta + \eta_i \quad \text{with} \quad \eta_i \sim_{i.i.d.} \mathcal{N}(0,\Gamma) \tag{4.2.2}$$

where β represents the vector of fixed effects to be estimated (i.e. unknown population parameters), X_i is a known design matrix, and η_i are individual random variables assumed i.i.d. $\mathcal{N}(0,\Gamma)$. We suppose that all the ε_{ij} and η_i are mutually independent.

4.2.2 REML version of the SAEM algorithm

REML estimation

Dempster et al. (1977) and Laird and Ware (1982) showed, in the context of linear mixed effects models, that REML estimates can easily be obtained from the EM algorithm. Indeed, using a Bayesian formulation of the model, the REML parameter estimates can be obtained by considering the fixed effects as part of the missing data vector with a normal distribution and a variance that tends to infinity. They are then automatically integrated with the other random effects using the EM algorithm.

We based the extension of this REML approach to nonlinear mixed effects models on a stochastic version of the EM algorithm, namely the Stochastic Approximation EM – SAEM (Delyon et al., 1999), as presented below. This algorithm indeed proved to be more computationally efficient than a classical Monte Carlo EM algorithm thanks to a recycling of the simulated variates from one iteration to the next.

SAEM-ML algorithm

Let $\theta = (\beta,\Gamma,\sigma^2)$ be the vector of parameters to be estimated ; $l(y;\theta)$ is the likelihood of the observed data $y = (y_{ij})_{(1\leq i\leq N, 1\leq j\leq n_i)}$ and $f(y,\phi;\theta)$ is the likelihood of the complete data $(y,\phi) = (y_{ij},\phi_i)_{(1\leq i\leq N, 1\leq j\leq n_i)}$. Thus, $l(y;\phi) = \int f(y,\phi;\theta)d\phi$. For the classical ML estimation, the goal is to compute the maximum likelihood estimator of the unknown set of parameters $\theta = (\beta,\Gamma,\sigma^2)$, by maximizing the likelihood of the observations $l(y;\theta)$.

The usual EM algorithm for linear mixed effects models is derived as follows. At iteration k, the E–step consists in computing the conditional expectation of the complete log–likelihood $Q_{k+1} = E(\log f(\boldsymbol{y},\boldsymbol{\phi};\boldsymbol{\theta})|\boldsymbol{y},\boldsymbol{\theta_k})$. The M–step computes the parameter values $\boldsymbol{\theta}_{k+1}$ that maximizes $Q_{k+1}(\boldsymbol{\theta})$. The EM sequence $(\boldsymbol{\theta}_{k+1})$ converges to a stationary point of the observed likelihood under general regularity conditions (Wu, 1983).

In the case of nonlinear mixed effects models, the expectation in the E step cannot be obtained analytically and has to be calculated using simulation. In order to reduce the amount of required simulations compared to a classical Monte Carlo EM algorithm (Wei and Tanner, 1990b), several authors (Delyon et al., 1999; Liao and Lipsitz, 2002; Levine and Casella, 2001) proposed to perform a 'recycling' of the simulated variates. At iteration k of the SAEM algorithm, the Expectation phase is replaced by the following stochastic approximation:

$$Q_{k+1}(\boldsymbol{\theta}) = Q_k(\boldsymbol{\theta}) + \gamma_{k+1}\left(\frac{1}{m}\sum_{\ell=1}^{m}\log\ f(\boldsymbol{y},\boldsymbol{\phi}^{[k+1,\ell]};\boldsymbol{\theta}) - Q_k(\boldsymbol{\theta})\right) \tag{4.2.3}$$

where vector $\boldsymbol{\theta}$ corresponds to $(\boldsymbol{\beta},\boldsymbol{\Gamma},\sigma^2)$, and the random vector $\boldsymbol{\phi}$ is simulated according to the conditional distribution $p(.|\boldsymbol{y};\boldsymbol{\theta}^{[k]})$, either directly or using a Hasting-Metropolis algorithm (Kuhn and Lavielle, 2004). Kuhn and Lavielle (2004) also showed that the convergence of the algorithm can be considerably improved by coupling it with an MCMC procedure, i.e. by simulating m Monte Carlo chains for vector $\boldsymbol{\phi}$ and averaging the obtained likelihood values over the m chains. Thanks to the 'recycling' process presented in the equation above and on the contrary to the classical Monte Carlo EM (MCEM) algorithm, the number of chains m does not have to be very large. Five chains is often sufficient in practice for ML estimates.

Parameter γ_k is a crucial parameter in this estimating procedure. It performs a smoothing of the calculated likelihood values from one iteration to the other and therefore considerably accelerates convergence compared to other MCMC estimation procedures. In practice, this smoothing parameter is defined as follows. During the first K iterations, γ_k=1, i.e. there is no smoothing performed and the algorithm is equivalent to an MCEM algorithm (Wei and Tanner, 1990b). McCulloch (1997) showed that this algorithm converged very rapidly towards a neighborhood of the ML estimates but then continued showing a great deal of variation. Therefore, from iteration $(K+1)$ the smoothing starts in order to stabilize the estimates and converge more rapidly towards the actual ML estimates (Kuhn and Lavielle, 2004). Parameter γ_k is a sequence of stepsizes within the interval [0,1]. It is recommended (Kuhn and Lavielle, 2004) to take $\gamma_k = (k-K)^{-1}$ for $k \geq (K+1)$. The choice of the iteration number K is crucial to obtain correct estimates because the

smoothing forces the algorithm to converge. Therefore, to ensure convergence towards the actual ML estimates, the user has to make sure the algorithm has already converged into a neighborhood of the MLEs. To do so, it is recommended to use this algorithm with several different starting values.

SAEM-REML algorithm

For the REML estimation, following Foulley and Quaas (1995), we considered the fixed effects as random, with a flat prior (i.e. $\pi(\boldsymbol{\beta})$ proportional to a constant). More specifically, we assumed vector $\boldsymbol{\beta}$ normally distributed with an infinite variance (Laird and Ware, 1982). The vector of parameters $\boldsymbol{\theta}$ to be estimated becomes $\boldsymbol{\theta}^\star = (\boldsymbol{\Gamma},\sigma^2)$, and the vector of random effects $\boldsymbol{\phi}$ now includes $\boldsymbol{\beta}$ and will be denoted $\boldsymbol{z} = (\boldsymbol{\eta},\boldsymbol{\beta})$, for the REML algorithm.

At iteration k, the SAEM-REML algorithm is therefore composed of the following steps:

- *Simulation–step*: draw $\boldsymbol{z}^{(k+1)}$ from the conditional distribution $p(\cdot|\boldsymbol{y};\boldsymbol{\theta}_k^\star)$.

- *Stochastic approximation–step*: update $Q_{k+1}(\boldsymbol{\theta}^\star)$ according to

$$Q_{k+1}(\boldsymbol{\theta}^\star) = Q_k(\boldsymbol{\theta}^\star) + \gamma_k \left(\frac{1}{m} \sum_{\ell=1}^{m} \log f(\boldsymbol{y},\boldsymbol{z}^{[k+1,\ell]};\boldsymbol{\theta}^\star) - Q_k(\boldsymbol{\theta}^\star) \right) \tag{4.2.4}$$

  where (γ_k) is a decreasing sequence of positive numbers, as presented above.

- *Maximization-step*: update $\boldsymbol{\theta}_k^\star$ according to

$$\boldsymbol{\theta}_{k+1}^\star = \text{Arg} \max_{\boldsymbol{\theta}^\star} Q_{k+1}(\boldsymbol{\theta}^\star). \tag{4.2.5}$$

Delyon et al. (1999) and Kuhn and Lavielle (2004) have shown that the SAEM algorithm converges to a maximum (local or global) of the likelihood of the observations under very general conditions.

In this context, the complete data of the model is $(\boldsymbol{y},\boldsymbol{z})$ and the complete log–likelihood

can be written as

$$\log f(\boldsymbol{y},\boldsymbol{z};\boldsymbol{\theta}^{\star}) = -\frac{N_{tot}+N\,d}{2}\log(2\pi) - \frac{N_{tot}}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i,j}(y_{ij}-g(\boldsymbol{t}_{ij},\boldsymbol{z}_i))^2$$
$$-\frac{N}{2}\log(|\boldsymbol{\Gamma}|) - \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{\eta}_i'\boldsymbol{\Gamma}^{-1}\boldsymbol{\eta}_i + const.$$

where $N_{tot} = \sum_{i=1}^{N} n_i$ is the total number of observations, N is the number of individuals and d is the dimension of vector $\boldsymbol{\phi}_i$ as defined in equation (4.2.2) for all individuals i.

Then, the approximation step reduces to updating the sufficient statistics of the complete model:

$$\tilde{\boldsymbol{s}}_{1,k+1} = \tilde{\boldsymbol{s}}_{1,k} + \gamma_{k+1}\left(\frac{1}{N}(\boldsymbol{\eta}^{(k+1)}\boldsymbol{\eta}^{(k+1)'}) - \tilde{\boldsymbol{s}}_{1,k}\right)$$
$$\tilde{\boldsymbol{s}}_{2,k+1} = \tilde{\boldsymbol{s}}_{2,k} + \gamma_{k+1}\left(\frac{1}{N_{tot}}\sum_{ij}(y_{ij}-g(\boldsymbol{\eta_i}^{(k+1)},\boldsymbol{\beta}^{(k+1)},\boldsymbol{t}_{ij}))^2 - \tilde{\boldsymbol{s}}_{2,k}\right)$$

and $\boldsymbol{\theta}^{\star}_{k+1}$ is obtained in the maximization step as follows:

$$\hat{\boldsymbol{\Gamma}}^{(k+1)} = \tilde{\boldsymbol{s}}_{1,k+1} \tag{4.2.6}$$
$$\hat{\sigma}^{2(k+1)} = \tilde{\boldsymbol{s}}_{2,k+1}. \tag{4.2.7}$$

When the simulation step cannot be directly performed, Kuhn and Lavielle (2004) proposed to combine this algorithm with a Markov Chain Monte Carlo (MCMC) procedure: the sequence $(\boldsymbol{z}^{(k)})$ is a Markov Chain with transition kernels $(\Pi_{\boldsymbol{\theta}^{\star}_k})$. Then, the simulation step becomes:

– *Simulation–step*: using $\boldsymbol{z}^{(k)}$, draw $\boldsymbol{z}^{(k+1)}$ from transition probability $\Pi_{\boldsymbol{\theta_k}^{\star}}(\boldsymbol{z}^{(k)})$.

To perform the simulation step, it is needed to know the conditional distribution $\boldsymbol{\beta},\boldsymbol{\eta}|\boldsymbol{y};\boldsymbol{\theta}^{\star}$ but it is sometimes simpler to use a Gibbs scheme, i.e. draw $\boldsymbol{\eta}^{(k+1)}$ from the conditional distribution $p(\cdot|\boldsymbol{y},\boldsymbol{\beta}^{(k)};\boldsymbol{\theta}^{\star}_k)$ and $\boldsymbol{\beta}^{(k+1)}$ from the conditional distribution $p(\cdot|\boldsymbol{y},\boldsymbol{\eta}^{(k+1)};\boldsymbol{\theta}^{\star}_k)$

In practice, we use m iterations of Hasting–Metropolis (H–M) algorithm to obtain an approximation of these conditional distributions. Then, to approximate, at iteration k,

the distribution of $\boldsymbol{\eta}^{(k+1)}|\boldsymbol{y},\boldsymbol{\beta}^{(k)};\boldsymbol{\theta}_k^\star$, three transition kernels were successively used:

- first, the prior distribution of $\boldsymbol{\eta_i}$ (for $i = 1,\cdots,N$), which is the Gaussian distribution
$$q^{(1)}_{\theta^\star_{i,k}} = \mathcal{N}(0,\Gamma_k);$$

- second, the random walk
$$q^{(2)}_{\theta^\star_{i,k}} = \mathcal{N}(\eta_{i,p-1},\rho_1^2\Gamma_k),$$
where ρ_1 is a constant and $p = 1,\cdots,h$;

- finally $q^{(3)}_{\theta^\star_{i,k}}$ is a succession of d unidimensional Gaussian random walks: each component of $\boldsymbol{\eta}$ are successively updated.

And to approximate the distribution of $\boldsymbol{\beta}^{(k+1)}|\boldsymbol{y},\boldsymbol{\eta}^{(k+1)};\boldsymbol{\theta}_k^\star$ two transition kernels were successively used:

- first, the random walk
$$q^{(4)}_{\theta^\star_{i,k}} = \mathcal{N}(\beta_{p-1},\rho_2^2\Gamma_{\beta^{(k)}}),$$
where ρ_2 is a constant and $\Gamma_{\beta^{(k)}}$ is an estimator of the posterior variance of $\boldsymbol{\beta}$;

- second, $q^{(5)}_{\theta^\star_{i,k}}$ is a succession of d unidimensional Gaussian random walks.

Then, the simulation–step at iteration k consists in running first h_1 iterations of the H–M algorithm with proposal $q^{(1)}_{\theta^\star_{i,k}}$, then h_2 iterations with proposal $q^{(2)}_{\theta^\star_{i,k}}$, h_3 iterations with proposal $q^{(3)}_{\theta^\star_{i,k}}$, h_4 iterations with proposal $q^{(4)}_{\theta^\star_{i,k}}$ and finally h_5 iterations with proposal $q^{(5)}_{\theta^\star_{i,k}}$.

The use of different kernels permits to increase the convergence and to favour all kind of transition. The values of parameters ρ_1, ρ_2, h_1, h_2, h_3, h_4 and h_5 involved in this simulation procedure have to be chosen by the user. Few iterations of the H–M algorithm at each simulation–step are enough to converge and in practice, h_1, h_2, h_3, h_4 and h_5 are less than 10. The choice of ρ_1 and ρ_2 is more delicate as they play an important role in the random walk. The values of ρ are linked to the acceptance rate of the H–M algorithm, so they must be chosen to approximate the 'optimal acceptance rate'. Some theoretical and empirical results (Roberts et al., 1997; Roberts and Rosenthal, 2001) have shown

that in high dimensions, under various regularity conditions, it is optimal to choose the scale parameter of the random walk such that the asymptotic acceptance rate of the H–M algorithm is approximately 0.234.

As suggested by Kuhn and Lavielle (2004), m Markov Chains were run to improve the convergence of the algorithm. The simulation step therefore required to draw m sequences $z^{(k+1,1)}, \ldots, z^{(k+1,m)}$ at iteration k and to combine stochastic approximation and Monte Carlo in the approximation step:

$$s_{k+1} = s_k + \gamma_k \left(\frac{1}{m} \sum_{l=1}^{m} \tilde{S}(y, z^{(k+1,l)}) - s_k \right). \tag{4.2.8}$$

Generally, it is necessary to use more chains in SAEM-REML than SAEM-ML, and in practice m is between 5 and 30.

4.3 Numerical examples

This algorithm was applied to several data sets in order to study its properties. Here, we present three applications of the estimation procedure. The first example is a linear mixed effects models, which allows to validate our algorithm with the standard EM algorithm which can be analytically implemented in the linear case. The second example is an analysis of real growth data in tree using an asymptotic regression model and it allows to compare our algorithm with R and SAS functions. Finally, in the last example we analyse a real growth data in a chicken using a nonlinear Gompertz function. To analyse the performance of our algorithm, we process a simulation study based on this data set.

4.3.1 A linear mixed model: ultrafiltration data

This data set was presented by Vonesh and Carter (1992). The data contain the ultrafiltration response of 20 membrane dialysers measured at 7 different transmembrane pressures with an evaluation made at 2 different blood flow rates. These data were analyzed in detail by standard second order algorithms in the SAS manual for mixed models (Littell et al., 1996) and with the EM algorithm by Foulley and van Dyk (2000).

It is a typical random coefficient model for longitudinal data analysis and can be

written as:

$$y_{ijk} = \mu + \alpha_i + \sum_{r=1}^{4} \beta_{ir} x_{ijk}^r + a_{ik} + \sum_{r=1}^{2} b_{r,ik} x_{ijk}^r + e_{ijk}, \tag{4.3.9}$$

where $y_{ijk} \in \mathbb{R}$ is the ultrafiltration rate in ml $\cdot$ h^{-1}, at the jth transmembrane pressure ($j = 1, \cdots, 7$), $\mu + \alpha_i$ the intercept for blood rate i ($i = 1,2$ for 200, 300 dl$\cdot$min^{-1}), $\sum_{r=1}^{4} \beta_{ir} x_{ijk}^r$ is the regression of the response on the transmembrane pressure x_{ijk} (dm Hg) as a homogeneous quadratic polynomial; a_{ik} and $b_{r,ik}$ represent the random coefficients up to the second degree of the regression defined at the dialyser level ($k = 1,2, \cdots, 20$).

Letting $\boldsymbol{\eta}_{ik} = (a_{ik}, b_{1,ik}, b_{2,ik})'$ and $\boldsymbol{e}_{ik} = \{e_{ijk}\}$, it is assumed that the individual random parameters $\boldsymbol{\eta}_{ik}$ are i.i.d. $\mathcal{N}(0,\Gamma)$ and the within-group errors $\boldsymbol{e}_{ik}$ are i.i.d. $\mathcal{N}(0,\sigma^2 \mathrm{I}_7)$. Here, the vector of population parameters is $\boldsymbol{\theta} = (\mu, \alpha, \beta, \Gamma, \sigma^2)$ for the ML method and $\boldsymbol{\theta}^\star = (\Gamma, \sigma^2)$ for the REML method.

As the model is linear, it is possible to compare the EM and the SAEM algorithms, using the ML and REML methods. For the REML method, we use the standard procedure based on Henderson's formula (see Foulley and van Dyk (2000) for more details) to perform these algorithms. In this example, for all the algorithms, we used 400 iterations. The stepsizes (γ_k) used with SAEM–ML and SAEM–REML is $\gamma_k = 1$ for $1 \leq k \leq 200$ and $\gamma_k = 1/(k - 200)$ for $201 \leq k \leq 400$. Estimations of the variance components are displayed in Table 4.1.

TAB. 4.1 – *Estimations of the variance components for the ultrafiltration response using the EM and SAEM algorithms with ML and REML methods.*

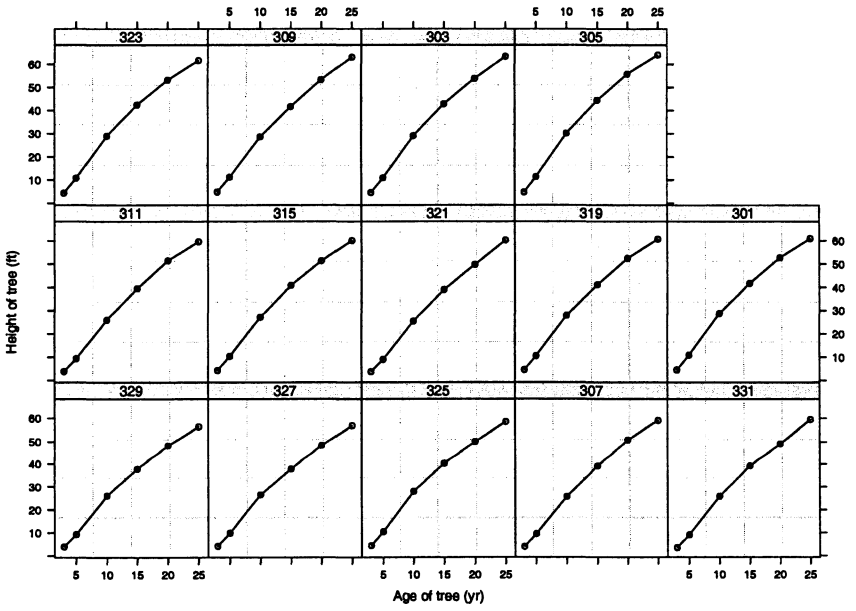
Method	Random Effects						Residual
	Γ_{00}	Γ_{01}	Γ_{02}	Γ_{11}	Γ_{12}	Γ_{22}	σ^2
EM-ML	1.79	-3.06	0.54	21.18	-6.00	1.91	3.15
EM-REML	2.25	-3.73	0.69	24.08	-6.83	2.17	3.32
SAEM-ML	1.82	-3.04	0.53	21.07	-5.97	1.90	3.15
SAEM-REML	2.23	-3.66	0.67	23.84	-6.76	2.15	3.32

As can be expected, there are no main differences between the EM and SAEM estimations, but the REML estimations, with both algorithms, appear to be larger than the ML estimations. This was to be expected as the ML estimates are known to be biased downwards.

4.3.2 A first nonlinear example: growth of Loblolly pine trees

These data, presented by Kung (1992), was studied by Pinheiro and Bates (2000) and consist in six measurements of the height of each of fourteen trees (Figure 4.1).

FIG. 4.1 – *Heights of Loblolly pine trees.*



Following Pinheiro and Bates (2000), an asymptotic regression model is used to explain the height y_{ij} of tree i at age x_j:

$$y_{ij} = \phi_{i1} + (\beta - \phi_{i1}) \exp[-\exp(\phi_{i2})x_j] + \varepsilon_{ij}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq n_i, \tag{4.3.10}$$

TAB. 4.2 – *Estimations with* **nlmixed** *(SAS),* **NLME** *(R) and* **SAEM** *(Matlab)*

	Method	Random Effects		Residual
NLMIXED	ML	7.840	0.001	0.479
	REML	-	-	-
NLME	ML	7.896	0.001	0.479
	REML	8.179	0.001	0.497
SAEM	ML	7.771	0.001	0.479
	REML	8.129	0.001	0.493

where $\phi_i = (\phi_{i1},\phi_{i2})'$ i.i.d. $\mathcal{N}(\mu,\Gamma)$, with

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \text{and} \quad \Gamma = \begin{pmatrix} \Gamma_{11} & 0 \\ 0 & \Gamma_{22} \end{pmatrix}.$$

In this example, we compare the performance of our algorithm with two others methods using the standard statistical softwares **SAS** and **R** to fit this model. These softwares have specific functions to fit nonlinear mixed–effects models. For example, the SAS Proc NLMIXED uses the quasi–Newton optimization technique and the adaptative Gaussian quadrature to obtain the ML estimates. The R library `NLME` (Lindstrom and Bates, 1990, Pinheiro and Bates, 2000 and The R project, http://www.r-project.org) allows to obtain the ML and the REML estimates using a first order linearization of the model.

The Table 4.2 gives the estimations of the variance components, using these three algorithms. For the SAEM algorithm, we used 1300 iterations for both methods and with 10 and 30 chains for the ML and REML methods, respectively. We used the following sequence (γ_k): $\gamma_k = 1$ for $1 \leq k \leq 500$ and $\gamma_k = 1/(k-500)$ for $501 \leq k \leq 1300$. The sequence of estimates using ML and REML methods are displayed in Figure 4.2 and Figure 4.3.

There are no main differences between the variance components estimates obtained with these three different algorithms but, again, the REML estimates are bigger than the ML estimates specially for Γ_{11}. The REML estimates obtained with our algorithm are very close to the estimates obtained with `NLME` what seems to validate our method.

FIG. 4.2 – *Evolution of ML estimates using SAEM. A logarithmic scale is used for the x-axis. Respectively, the fixed effects (μ_1,β,μ_2), the variances of random effects (Γ_{11},Γ_{22}) and the variance of the error (σ).*

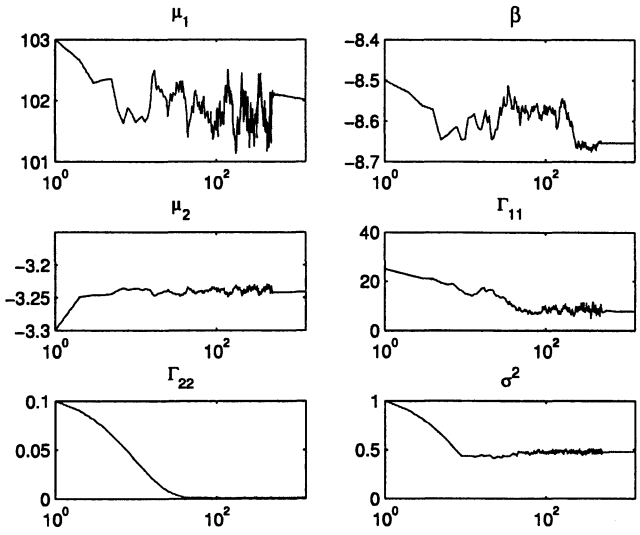
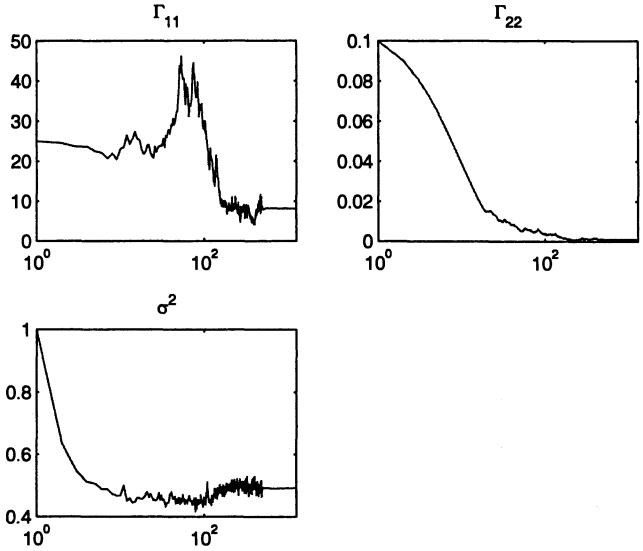


FIG. 4.3 – *Evolution of REML estimates using SAEM. A logarithmic scale is used for the x-axis. Respectively, the variance of random effects* (Γ_{11},Γ_{22}) *and the variance of the error* (σ).



4.3.3 A genetic example: growth curve analysis in chicken

These data come from a genetic selection experiment on the form of the growth curve initiated by F. Ricard in 1960 on meat-type chickens, and were analyzed by Mignon-Grasteau et al. (2000) using Bayesian methods. There were five lines of selection. Line X+− was selected for high juvenile body weight at 8 weeks and low adult body weight at 36 weeks. In contrast, line X−+ was selected for low juvenile body weight and high adult body weight. In line X++, animals were selected for high body weights at both ages and, in the opposite line, X−−, they were selected for low body weights at both ages. Line X00 was an unselected control line.

The data set analyzed here comes from the last generation of selection and comprised 10 animals from each line, i.e. a total of 50 animals, with 11 growth measurements for each animal at ages 4, 6, 8, 12, 16, 20, 24, 28, 32, 36 and 40 weeks.

The Gompertz function was used to model these growth curves and the model can be written as follows (for animal i, in line l, at time t_j):

$$y_{lij} = A_{li} \exp(-\beta \exp(-C_{li} t_j/100)) + \varepsilon_{lij} \tag{4.3.11}$$

where A_{li} is the asymptotic body weight of animal i (in selection line l), i.e. the weight at an infinite age. Parameter C_{li} corresponds to the maturation rate, i.e. the rate at which the animal approaches its asymptotic weight. Due to convergence problems, parameter β was considered as a fixed effect, i.e. with the same value for all the animals.

The two parameters of the curve A_{li} and C_{li} were assumed normally distributed and correlated. The different lines were fitted as fixed effects, as follows:

$$A_{li} = \alpha_l + a_{li}, \quad \text{where } a_{li} \sim_{i.i.d.} \mathcal{N}(0,\Gamma_1) \tag{4.3.12}$$

$$C_{li} = \gamma_l + c_{li}, \quad \text{where } c_{li} \sim_{i.i.d.} \mathcal{N}(0,\Gamma_2). \tag{4.3.13}$$

Furthermore, it is assumed that $cov(a_{li},c_{li}) = \Gamma_{ac}$. The residuals ε_{lij} were also assumed *i.i.d.*, normally distributed with mean zero and constant variance σ^2_ε.

These data were analyzed using the SAEM-REML algorithm presented above. Then, in order to check the gain obtained with the REML methodology compared to ML in non-linear mixed effects models on variance component estimations, we simulated 500 data sets using the same data structure and the previously obtained parameter values.

Data were simulated using the following parameters:
$\boldsymbol{\alpha} = \{\alpha_l; l = 1,\cdots,5\} = (3050,1930,3080,1780,2390)'$, $\beta = 4.30$, $\boldsymbol{\gamma} = \{\gamma_l; l = 1,\cdots,5\} =$

(13.3,17.9,17.1,15.8,15.1)', $\Gamma_1 = 56100$, $\Gamma_2 = 1.61$ and $\Gamma_{ac} = -90.16$ ($\rho_{ac} = -0.30$), $\sigma_\epsilon^2 = 8000$.

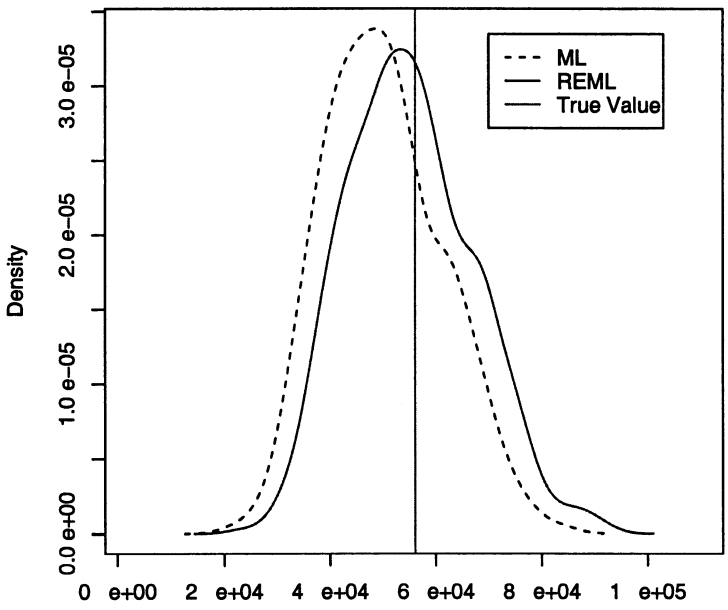
The same initial values were used for both ML and REML, which were: $\alpha_0 = (4000,2000,4000,2000,3000)'$, $\beta = 10$, $\gamma_0 = (15,20,20,15,15)'$, $\Gamma_1 = 100000$, $\Gamma_2 = 5$ and $\rho_{ac} = 0$, $\sigma_\epsilon^2 = 60000$. The SAEM algorithm was found to be very robust to the choice of the initial values for the variance parameters. In this example, we used 2000 iterations, and the following sequence (γ_k): $\gamma_k = 1$ for $1 \leq k \leq 1500$ and $\gamma_k = 1/(k - 1500)$ for $1501 \leq k \leq 2000$.

Summary statistics for ML and REML estimates obtained for these 500 simulated data sets are given in Table 4.3. The true values of the parameters used in the simulation, the means and individual confidence intervals (at a 95% level) of the sampling distributions of the variance components estimates and the estimated relative mean squared errors of the estimates are provided. It can be seen that the mean values for the REML estimates were closer to the simulated values for parameters Γ_1 and Γ_2. A test was also performed showing that the differences observed between the two estimation procedures were significant (at a 5% level) for these two variance parameters. Moreover, the individual confidence intervals of REML estimates of these parameters, at a 95% level, include the true value for these parameters on the contrary to the ML estimates showing that the SAEM-REML algorithm was able to correct the bias observed with ML. Figure 4.4 give a graphical representation of these results showing the density estimates obtained with ML and REML, for the variance parameter Γ_1 and for the balanced case.

TAB. 4.3 – *Summary statistics from the complete data sets.*

	Method	Γ_1	Γ_2	σ^2	ρ_{ac}
True value		56100	1.61	8000	-0.30
Mean	ML	50058	1.3922	8006.3	-0.29495
	REML	55091	1.5807	8038.5	-0.28842
Individual	ML	[49081;51034]	[1.356;1.43]	[7958.4;8054.3]	[-0.309;-0.280]
I.C. 95%	REML	[54051;56130]	[1.54;1.62]	[7990.1;8086.9]	[-0.302;-0.275]
RMSE	ML	22.532	29.028	6.818	55.234
%	REML	21.148	31.233	6.895	49.671
p-value		7.4134e-012	1.4817e-010	0.35	0.51

FIG. 4.4 – *Complete data sets. The density estimates obtained with ML and REML for* Γ_1.



On the other hand, although the means seem to be slightly closer to the simulated values for the residual variance and the correlation parameters with ML rather than REML, the estimations were not found to be significantly different. In general, the REML estimates were found to be slightly more variable than the ML estimates, except for the correlation parameter. It is expected that even larger differences would be observed between ML and REML when more fixed effects are included in the model.

REML methodology is also known, in linear models, to be more robust than ML for parameter estimations in unbalanced data sets. In order to investigate its properties in the nonlinear mixed model framework, two drop-out processes (as defined by Little and Rubin, 1987) were used on the previously simulated data sets: MAR (missing at random) and MCAR (missing completely at random). In both cases, 40% of the observations were deleted, i.e. 4 animals per line were chosen and measurements after 20 weeks were deleted. In the MAR case, deletion of the measurements were made in each line for the 4 animals with the lowest individual a_{li} prediction at 20 weeks. 500 data sets were then simulated with these two drop-out structures and were analyzed with both ML and REML.

As for the complete data sets, we used 2000 iterations for both algorithms and the same stepsizes (γ_k). Results are presented in Tables 4.4 and 4.5. They were found to be very similar for both drop-out processes. REML proved to performed even better than ML in these unbalanced cases than for the complete data. Indeed, the relative mean squared error (RMSE) values were found smaller with the REML than with ML for all variance-covariance parameters. As for the complete data, differences between ML and REML were found significant only for the two variance parameters Γ_1 and Γ_2. The mean value for the correlation parameter was now found closer to the simulated value with REML than with ML in the unbalanced cases.

In this example, the CPU time required for one iteration of SAEM–ML and SAEM–REML is different due to the simulation step which is more complex for the REML estimation. However, this time difference is not important. We measured these CPU time for one data set with a computer using a AMD bi–processor Opteron 248 at 2.26GHz. Then, considering m = 5, the number of simulations at each iteration in the simulation step for both algorithm, SAEM–ML spends 24 milliseconds per iteration and SAEM–REML spends 35 milliseconds. When m = 20, SAEM–ML spends 104 milliseconds per iteration and SAEM–REML 148 milliseconds.

We tried to compare this estimation procedure on the 500 simulated data sets with the REML version of the `nlme` function of R. Convergence problems were, however, encountered for a large number of these simulated data sets, for both the balanced and unbalanced

cases, which made these comparisons very difficult. On the contrary, the SAEM algorithm converged for all the data sets studied here.

TAB. 4.4 – *Summary statistics from the unbalanced data sets obtained by the MAR procedure.*

	Method	Γ_1	Γ_2	σ^2_ϵ	ρ_{ac}
True value		56100	1.61	8000	-0.30
Mean	ML	49371	1.3108	8039.8	-0.26213
	REML	54332	1.5253	8041	-0.26651
Individual	ML	[48228;50514]	[1.266;1.356]	[7982.6;8097.1]	[-0.285;-0.240]
I.C. 95%	REML	[53123;55540]	[1.481;1.570]	[7987.8;8094.2]	[-0.281;-0.252]
RMSE	ML	26.083	36.760	8.156	86.256
%	REML	24.689	31.791	7.576	57.206
p-value		6.26e-009	4.15e-011	0.98	0.75

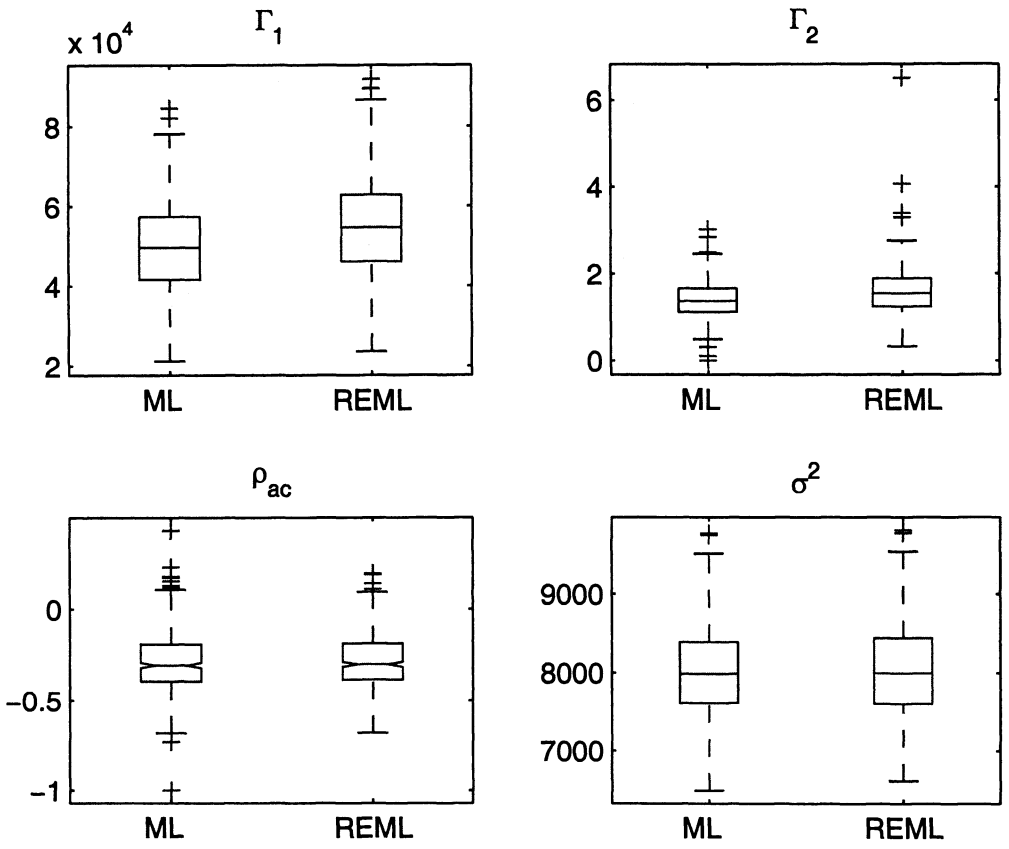
TAB. 4.5 – *Summary statistics from the unbalanced data sets obtained by the MCAR procedure.*

	Method	Γ_1	Γ_2	σ^2_ϵ	ρ_{ac}
True value		56100	1.61	8000	-0.30
Mean	ML	49505	1.3194	8042.1	-0.26255
	REML	54806	1.5376	8044.3	-0.2676
Individual	ML	[48359;50651]	[1.273;1.366]	[7985.0;8099.2]	[-0.285;-0.240]
I.C. 95%	REML	[53577;56036]	[1.492;1.583]	[7991.5;8097.1]	[-0.283;-0.252]
RMSE	ML	26.034	37.49	8.1337	86.447
%	REML	25.024	32.219	7.529	58.508
p-value		8.4356e-010	6.207e-011	0.96	0.72

Figures 4.5 to 4.7 give a graphical representation of these results showing the boxplot of the variance components estimates obtained with ML and REML, for the three data

sets studied here, i.e. the complete data and the unbalanced data sets obtained by the MAR and the MCAR procedures.

FIG. 4.5 – *Boxplot of variance components estimates with balanced data sets.*



4.4 Discussion

The aim of this work was to present a novel estimation procedure to obtain REML parameter estimates in nonlinear mixed effects models. The proposed approach is based

FIG. 4.6 – *Boxplot of variance components estimates with unbalanced data sets obtained with the MAR procedure.*

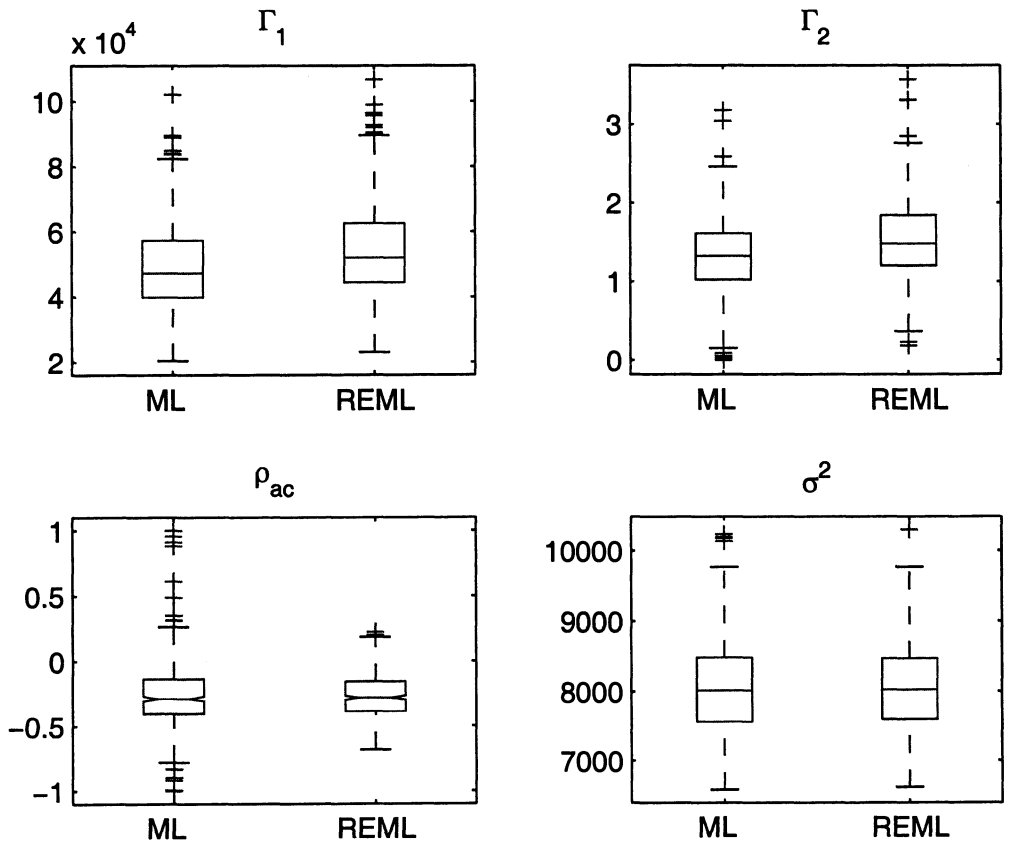
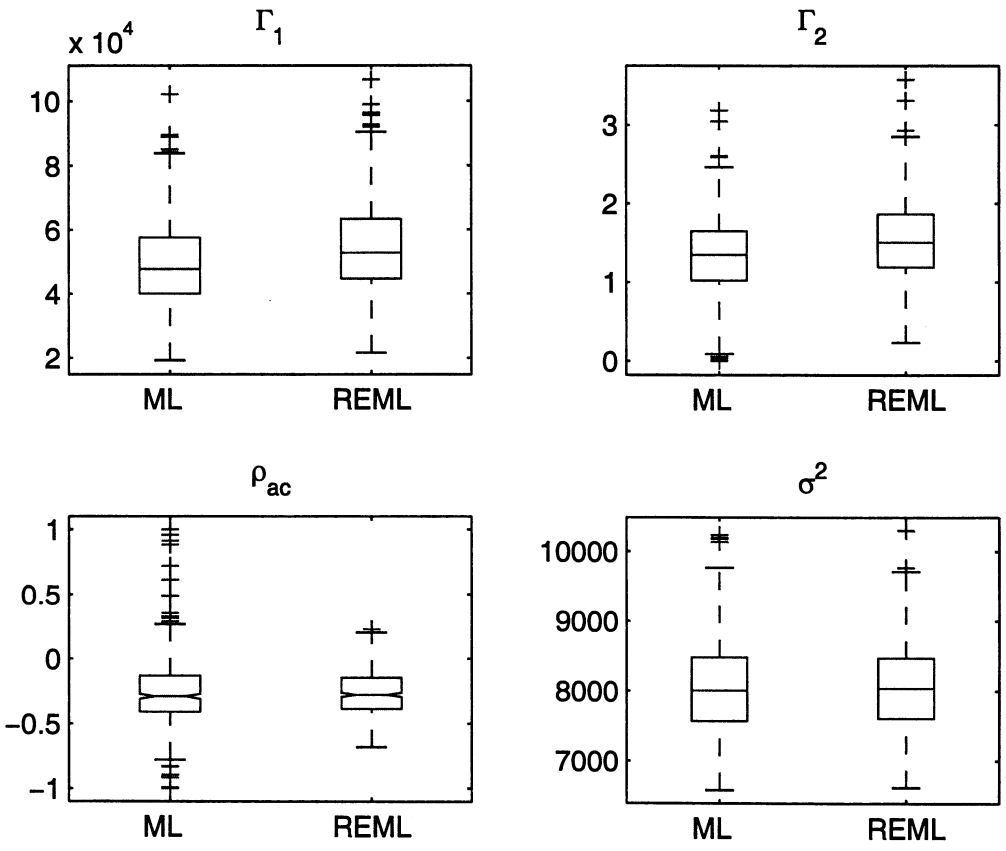


FIG. 4.7 – *Boxplot of variance components estimates with unbalanced data sets obtained with the MCAR procedure.*



on an integration of the fixed effects using a stochastic EM algorithm.

The stochastic version of the EM algorithm used here was the Stochastic approximation EM (SAEM) algorithm originally proposed by Delyon et al. (1999). It is conceptually very simple and has several advantages compared to the classical Monte Carlo EM algorithm (Wei and Tanner, 1990b). Firstly, thanks to the 'recycling' of the simulated values from one iteration to the next it allows to considerably reduce the number of Monte Carlo chains required. Secondly, the smoothing parameter considerably accelerates convergence to the MLEs. Comparison of the SAEM algorithm with approximated estimation procedures such as First Order Conditional Estimation (FOCE), Laplacian methods or the Gaussian quadrature (see Davidian and Giltinan, 2003) was performed by Kuhn and Lavielle (2004). The SAEM algorithm was found to perform better than the other methods in terms of robustness with regard to the choice of the starting values, especially for the variance components, and accuracy of the estimates. It is also much faster to converge than classical Bayesian methods such as the Gibbs sampling.

Although ML is still most often used for estimation in nonlinear mixed effects models in various biometrical fields, such as pharmacokinetic research, it provides, as in the linear case, bias estimates of the variance-covariance components. As shown in the simulation study, the proposed REML estimation procedure was able to adequately correct this bias. This was especially the case for the variance components of the growth curve parameters. Less differences were observed between ML and REML for the residual variance.

It was also observed that the REML estimation procedure was more robust than ML for unbalanced data sets. Indeed, the relative mean squared errors (RMSE) were found to be smaller with REML for all the variance-covariance parameters for unbalanced data sets. On the other hand, nature of the drop-out process (MAR or MCAR) did not seem to affect the properties of the REML estimates. In this study, five genetic lines were considered as fixed effects. As pointed out by Verbeke and Molenberghs (2000), differences between ML and REML estimates would be even larger when more fixed effects are included in the model.

The marginalization process employed here to derive the REML estimation of variance covariance components precludes, however, any inference on the fixed effects as we get rid of these effects by integrating them out. Nevertheless, their estimation in a non standard maximum likelihood context remains an important issue. For example, Gianola and Sorensen (2006) recently proposed a new class of estimators of fixed effects in linear mixed models based on the concept of integrated likelihood but here with respect to the variance components. The differences observed in terms of MSE between this new estimator (called

LIFE) and the traditional ones remain, however, rather small in the examples tested and simulations considered. What we can do presently is to suggest some possible estimator of fixed effects obtained directly as a by product of the SAEM-REML algorithm. An obvious one is an equivalent of the so called "Estimated GLS" for the linear model consisting here of the expectation of the conditional distribution of fixed effects given the data vector and the variance covariance equal to their REML estimators. This estimator makes sense in an Empirical Bayesian framework. But its frequentist properties require a study per se in comparison with other potential estimators.

This SAEM-REML algorithm was proposed here in the context of nonlinear mixed effects models, where all the effects were asssumed normally distributed. It can also be extended for REML estimation in generalized linear mixed effects models (GLMM), for the analysis of repeated categorical traits. This would be especially useful in epidemiological studies for the analysis of disease resistance characters. Furthermore, it would be interesting to compare, in the GLMM framework, this algorithm with existing ML procedures (Gueorguieva and Agresti, 2001) and with the 'REML' approach proposed by Liao and Lipsitz (2002). It is expected that our method would be less computationally demanding than the latter as fewer simulations are needed.

It is important to note that we tried to use the standard R function `nlme` in the simulation study but found several convergence problems which prevented us to compare it whith our algorithm.

Lavielle and Meza (2006), proposed a 'parameter expansion' version of the SAEM algorithm, namely the PX-SAEM algorithm, by adapting the PX-EM algorithm proposed by Liu et al. (1998) to stochastic methods and nonlinear mixed effects models. This PX-SAEM was able to considerably improve the speed of convergence of the algorithm, especially for the first EM iterations. It is expected that this PX extension would also be useful to reduce the computing time required for the SAEM-REML estimation.

Chapitre 5

Application of SAEM in Generalized Linear Mixed Models: the Probit Model

Summary

Generalized linear mixed models (GLMM) form a very general class of random effects models for discrete and continuous responses in the exponential family. They are useful in a variety of applications. The traditional likelihood approach for GLMM usually involves high dimensional integrations which are computationally intensive. In this chapter, we study the case of binary outcomes and the probit model. We show that maximum likelihood estimation can be achieved in this model by using the SAEM algorithm. It appears to be less sensitive to the choice of starting values than classical methods based on approximation technics or numerical approximations. Furthermore, we adapt the PX–SAEM algorithm to the GLMM obtaining a significant improvement of the convergence of SAEM. We propose also a strategy to obtain REML estimates for the variance components in this kind of models using the SAEM algorithm.

Contents

5.1 Introduction

Generalized linear mixed models (GLMM) (see Breslow and Clayton, 1993) are natural extensions of the generalized linear model (GLM) for analyzing non gaussian data collected from different clusters or from longitudinal studies. They incorporate random effects into the linear predictors, and include the well known linear mixed models for normal responses (Laird and Ware, 1982) as a special case. They provide a flexible likelihood framework under which population characteristics can be modeled as fixed effects and individual variations can be modeled as random effects.

GLMM apply to either continuous or discrete data. Regarding the latter, a popular class of GLMM is the probit–normal model which permits to study as well the binary data as the ordinal data. The models for binary response variable are important in many

fields of research, since subjects are often classified in two categories. A convenient way to model such data consists of discretizing with a threshold a latent continuous distribution, a process that has been used extensively in the biometrics and econometrics literature (see Ashford and Sowden, 1970; McFadden, 1989; Hausman and Wise, 1978) and is especially useful for joint modeling of continuous and discrete outcomes (Gueorguieva and Agresti, 2001).
Two main choices prevail with the respect to the distribution of the latent continuous variable, i.e. either a logistic or a standard normal. The differences between the two involve technical aspects (sufficient statistics in the case of logit) but also interpretation properties in relationship with their practical domains of applications. As the logit model relies on the concept of "odds" (this is a linear model on a log odd scale), it was specially well fitted to deal with applications in heath sciences (epidemiology, clinical trials) involving estimation of relative risks or benefits. The probit model emerged from areas deeply noted in theories laying on the Bell shape curves such as quantitative genetics and inheritance issues (see Wrights's works, Falconer, 1989, Foulley and Manfredi, 1991 and Gianola, 1982). In that context, parameters on the latent scale are interpreted as those of usual continuous traits and this is specialty convenient for joint modeling of continuous and binary outcomes (Foulley et al., 1983; Gueorguieva and Agresti, 2001).

The estimation of the parameters in the GLMM is difficult to compute because the exact likelihood function involves an intractable high–dimensional integration. Therefore, several approximations to the likelihood function and approximate maximum likelihood estimators (MLE) have been proposed in the previous literature (Schall, 1991; Breslow and Clayton, 1993; Wolfinger, 1992). Many of these approaches are reviewed in McCulloch (1997) and Rodríguez and Goldman (1995). The most frequently used methods are based on first or second order Taylor expansion. Among them, the penalized quasi-likelihood (PQL) by Breslow and Clayton (1993) is one of the most popular for the GLMM. It approximates the high-dimensional integration using the well–known Laplace approximation. Several authors (Breslow and Lin, 1995; Rodríguez and Goldman, 1995; Raudenbush et al., 2000) have reported downwardly biased estimates using these procedures in certain situations.
Alternatively, numerical integration can be used to perform the integration over the random–effects distribution. Gauss–Hermite quadrature can be used to approximate the above integral to any practical degree of accuracy. The Gauss–Hermite quadrature have been implemented in main software like MIXOR (Hedeker and Gibbons, 1996), MIXNO (Hedeker, 1999) and SAS PROC NLMIXED.

Others methods can be used to approximate the integration over the random effects distribution such as the Markov chain Monte Carlo (MCMC) procedure implemented via for example the Gibbs Sampling. These methods are, however, extremely time consuming.

Though the fact that these methods are slower than quadrature solution, Gibbs sampling may be more advantageous for models with many random effects.

In this chapter, we propose a new alternative method to estimate the parameter in the mixed–effects probit model within an exact estimation scheme using a stochastic EM algorithm (SAEM). We show so that it can be possible to improve convergence toward the maximum likelihood estimate using PX–SAEM within this particular context. Furthermore we propose a novel REML estimation procedure for these models, based on an integration of the fixed effects to reduce the bias observed with the ML estimation of variance component (see Harville, 1974).

In Section 5.2, we introduce the GLMM in a general way before to present, in Section 5.3 the mixed–effects probit model for dichotomous outcomes, used in this work. Then, the SAEM algorithm is specify to obtain the Maximum Likelihood and Restricted Maximum Likelihood estimates for this kind of models. The parameter expansion version of SAEM is introduced in Section 5.5. Some numerical experiments are proposed in Section 5.7.

5.2 The Generalized Linear Mixed Model

The generalized linear mixed models are natural extensions of generalized linear models (GLM) therefore we introduce first the GLM.

The GLM generalize the standard linear models in term of probability and linear link. They allow to analyse as well discrete data as continuous data. Following McCullagh and Nelder (1989), three hypotheses allow to characterize a GLM: a random component (distribution of the response vector), a systematic component (the linear function of the covariates) and a link function.

- **The distribution of the response vector**

In the context of the repeated data, the elements of the response vector $\boldsymbol{y_i} = \{y_{ij}\}$, with $i = 1, \cdots, N$ and $j = 1, \cdots, n_i$, are assumed to be independent with a distribution in the exponential family (see Nelder and Lee, 1992):

$$f(\boldsymbol{y_i}; \theta_i, \psi_i) = \exp\left\{ \frac{\boldsymbol{y_i}\theta_i - b(\theta_i)}{a(\psi_i)} + c(\boldsymbol{y_i}, \psi_i) \right\} \tag{5.2.1}$$

for some specified functions $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$ with $a(\psi_i) = \frac{\psi}{w_i}$, where ψ is called a dispersion parameter and w_i are known weights. Each member of the exponential family is

specify by the mean $\mu_i = b'(\theta_i)$ and the variance $\nu(\mu_i) = b''(\theta_i)a(\psi_i)$. The most important distribution of the form (5.2.1) are the normal, the bernoulli, the binomial, the Poisson and the gamma distributions.

- **The linear function of the covariates**

The systematic component is a linear function of the covariates where the linear predictor is as follows

$$\boldsymbol{\omega_i} = \boldsymbol{X_i}\boldsymbol{\beta},$$

where $\boldsymbol{X_i}$ is a $n_i \times p$ known matrix and $\boldsymbol{\beta}$ is a $p \times 1$ unknown vector.

- **The link function**

The link function $g(\cdot)$ is a strictly monotonic differentiable function which relates the expected value of the response distribution $\boldsymbol{\mu_i}$ to the linear predictor $\boldsymbol{\omega_i}$:

$$\boldsymbol{\omega_i} = g(\boldsymbol{\mu_i}).$$

If the response distribution is normal and $g(\boldsymbol{\mu}) = \boldsymbol{\mu}$, we recover the usual linear regression model.

For the Bernoulli and binomial distributions we have $0 < \mu_{ij} < 1$ and a link should satisfy the condition that it maps the interval (0,1) on to the whole real line. The four principal link functions associated to these distributions are:

1. *the logit or logistic function*

$$\omega_{ij} = \log\{\mu_{ij}/(1 - \mu_{ij})\};$$

2. *the probit or inverse Normal function*

$$\omega_{ij} = \Phi^{-1}(\mu_{ij})$$

where $\Phi(\cdot)$ is the Normal cumulative distribution function;

3. *the complementary log–log function*

$$\omega_{ij} = \log\{-\log(1 - \mu_{ij})\};$$

4. *the log–log function*

$$\omega_{ij} = \log\{-\log(\mu_{ij})\}.$$

To introduce random effects in GLM, we can consider GLMM as a two–stage model:

1. at the first stage, the observations y_{ij} are treated as conditionally independent (given the random effects $\boldsymbol{\eta_i}$) and described by a standard GLM with linear predictor:

$$\boldsymbol{\omega_i} = \boldsymbol{X_i\beta} + \boldsymbol{Z_i\eta_i} \tag{5.2.2}$$

   where $\boldsymbol{X_i\beta}$ is defined as previously and $\boldsymbol{Z_i}$ is a $(n_i \times d)$ incidence matrix pertaining to the random effects;

2. at the second stage, the individual random effects $\boldsymbol{\eta_i}$ are assumed to be i.i.d. with $\boldsymbol{\eta_i} \sim \mathcal{N}(0,\Gamma)$.

An important feature of such mixed–effects models is their difference with marginal models contrary to what happens with linear model.

In a linear model, the expectation of the observations is given as

$$\begin{aligned} E(y_{ij}) &= E_{\boldsymbol{\eta_i}}(E(y_{ij}|\boldsymbol{\eta_i})) \\ &= E_{\boldsymbol{\eta_i}}(X_{ij}\boldsymbol{\beta} + Z_{ij}\boldsymbol{\eta_i}) \\ &= X_{ij}\boldsymbol{\beta} \end{aligned}$$

where, in a nonlinear model as here,

$$\begin{aligned} E(y_{ij}) &= E_{\boldsymbol{\eta_i}}[g(X_{ij}\boldsymbol{\beta} + Z_{ij}\boldsymbol{\eta_i})] \\ &\neq g(X_{ij}\boldsymbol{\beta}). \end{aligned}$$

This shows as clearly pointed out by Molenberghs and Verbeke (2005), that the regression parameters have a complete different meaning in marginal and random effects models thus sometimes referred as "population average" and "subject–specific" models respectively.

The estimation of GLMM using ML or REML is not so straightforward, because the likelihood function includes integrals that are analytically intractable. There are two ge-

neral practical approaches to estimating GLMMs:

- Evaluate the likelihood using some method of numerical integration, such as quadrature (for simple cases) or simulation (for more complicated ones).

- Use a method that approximates the maximum-likelihood estimate.

5.3 The probit normal model for dichotomous outcomes

Probit regression (Finney, 1971) for dichotomous outcomes is popular in many fields, as genetic, epidemiology, sociological and econometrics studies. When the binary responses are clustered, for example repeatedly measured within individuals, or clustered within any other strata, the use of mixed model is necessary.

Let y_{ij} be the outcome of a dichotomous variable (coded as 0 or 1), associated with the jth observation of the individual i, $1 \leq i \leq N$, $1 \leq j \leq n_i$. The probability p_{ij} of a positif event, i.e. $y_{ij} = 1$, can be expressed in terms of the standard normal cumulative distribution function (probit model) but also the logistic cumulative distribution function (logit model). In this work we decide to focus on the probit model but the extensions of our results to other link functions are discuted in Section 5.8.

In the GLMM context, the probit model uses the probit link:

$$\begin{aligned} y_{ij} &\sim \text{Ber}(p_{ij}), \quad \text{with} \\ p_{ij} &= P(y_{ij} = 1) \\ &= \Phi(X_{ij}\beta + Z_{ij}\eta_i) \end{aligned} \tag{5.3.3}$$

$$\Rightarrow \Phi^{-1}(p_{ij}) = X_{ij}\beta + Z_{ij}\eta_i, \quad 1 \leq i \leq N, \quad 1 \leq j \leq n_i$$

where $\eta_i \in \mathbb{R}^d$ follows a Gaussian distribution $\mathcal{N}(0,\Gamma)$ and $\Phi(\cdot)$ denotes the normal cumulative distribution function. X_{ij} and Z_{ij} are two p and d known vectors.

In terms of the underlying latent ω, the model in (5.3.3) is written as:

$$\begin{aligned} y_{ij} &= sign(\omega_{ij}) = \begin{cases} 1 \text{ if } \omega_{ij} > 0 \\ 0 \text{ si } \omega_{ij} \leq 0 \end{cases}, \text{ with} \\ \omega_{ij} &= X_{ij}\beta + Z_{ij}\eta_i + \varepsilon_{ij}, \end{aligned} \tag{5.3.4}$$

where $\varepsilon_{ij} \sim \mathcal{N}(0,1)$ and where $(\boldsymbol{\eta_i})$ and (ε_{ij}) are assumed to be mutually independent.

Our purpose is then to compute the maximum likelihood (ML) and restricted maximum likelihood (REML) estimates of the respective unknown parameter vectors $\boldsymbol{\theta} = (\boldsymbol{\beta},\boldsymbol{\Gamma})$ and $\boldsymbol{\theta}^\star = (\boldsymbol{\Gamma})$.

5.4 SAEM–ML estimation for dichotomous outcomes models

The Stochastic Approximation EM (Delyon et al., 1999) was used here to obtain the ML estimates in the context of mixed–effects probit model for dichotomous outcomes. This algorithm proved to be more computationally efficient than a classical Monte Carlo EM algorithm thanks to a recycling of the simulated variates from one iteration to the next.

We consider here that the non observed data are $\boldsymbol{z} = (\boldsymbol{\omega},\boldsymbol{\eta})$, where $\boldsymbol{\omega} = (\omega_{ij}, 1 \leq i \leq N, 1 \leq j \leq n_i)$ is the vector of non observed latent variables, $\boldsymbol{\eta} = (\boldsymbol{\eta_i}, 1 \leq i \leq N)$ is the vector of random effects, and $\boldsymbol{y} = (y_{ij}, 1 \leq i \leq N, 1 \leq j \leq n_i)$ are the observed data. Then, the completed data are $(\boldsymbol{y},\boldsymbol{z}) = (\boldsymbol{y},\boldsymbol{\omega},\boldsymbol{\eta})$.

The complete data log likelihood is:

$$l(\boldsymbol{\theta}) = \log f(\boldsymbol{y},\boldsymbol{\omega},\boldsymbol{\eta};\boldsymbol{\theta}) = \log p(\boldsymbol{y}|\boldsymbol{\omega},\boldsymbol{\eta};\boldsymbol{\theta}) + \log p(\boldsymbol{\omega}|\boldsymbol{\eta};\boldsymbol{\theta}) + \log p(\boldsymbol{\eta};\boldsymbol{\theta})$$

where $\boldsymbol{\omega_i}|\boldsymbol{\eta};\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{X_i\beta} + \boldsymbol{Z_i\eta_i}, I_{n_i})$ and $\boldsymbol{\eta_i};\boldsymbol{\theta} \sim \mathcal{N}(0,\boldsymbol{\Gamma})$.

Notice that the first term does not bring any information about the parameters since $\boldsymbol{y}$ is completely specified without any uncertainty as long as $\boldsymbol{\omega}$ is known, it can be ignored. Then, we have:

$$\begin{aligned} l(\boldsymbol{\theta}) &= -\frac{N_{tot}}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{n_i}(\omega_{ij} - X_{ij}\boldsymbol{\beta} - Z_{ij}\boldsymbol{\eta_i})^2 \\ &- \frac{N}{2}\log 2\pi - \frac{N}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{\eta_i}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\eta_i}, \end{aligned} \tag{5.4.5}$$

where $N_{tot} = \sum_{i=1}^{N} n_i$ is the total number of observations.

Equation (5.4.5) allows us to choose the following minimal sufficient statistics:

$$\tilde{S} = \left(\tilde{S}_1, \tilde{S}_2\right) \tag{5.4.6}$$

with

$$\tilde{S}_1 = \sum_{i=1}^{N} \sum_{j=1}^{n_i} X'_{ij}(\omega_{ij} - Z_{ij}\eta_i) \tag{5.4.7}$$

$$\tilde{S}_2 = \sum_{i=1}^{N} \eta_i \eta_i{}'. \tag{5.4.8}$$

Then in this context, at iteration k, the SAEM algorithm is composed of the following steps:

- *Simulation–Step:* draw η and ω from the joint distribution $\omega,\eta|y;\theta_k$. We use for this a Gibbs scheme:

$$\begin{aligned} \eta_i{}^{(k+1)}|\omega^{(k)},y;\theta_k &\sim \mathcal{N}(W_i{}^{(k)} Z_i{}'(\omega_i{}^{(k)} - X_i\beta_k), W_i{}^{(k)}), \\ &\quad \text{where } W_i{}^{(k)} = [Z_i' Z_i + \Gamma_k{}^{-1}]^{-1} \\ \omega_{ij}^{(k+1)}|\eta^{(k+1)},y;\theta_k &\sim \mathcal{N}_{\pm}\left(X_{ij}\beta_k + Z_{ij}\eta_i{}^{(k+1)},0,1\right) \\ &\quad \text{the truncated normal distribution in 0} \\ &\quad \text{so that } sign(\omega_{ij}) = y_{ij}. \end{aligned}$$

- *Stochastic Approximation–Step:* the sufficient statistical are compute as follows:

$$\begin{aligned} s_{1,k+1} &= s_{1,k} + \gamma_k \left(\tilde{S}_{1,k+1} - s_{1,k}\right) \\ s_{2,k+1} &= s_{2,k} + \gamma_k \left(\tilde{S}_{2,k+1} - s_{2,k}\right). \end{aligned}$$

- *Maximization–Step:* update θ_k:

$$\begin{aligned} \beta_{k+1} &= \left(\sum_{i=1}^{N} \sum_{j=1}^{n_i} X'_{ij} X_{ij}\right)^{-1} s_{1,k+1}, \\ \Gamma_{k+1} &= \frac{s_{2,k+1}}{N}. \end{aligned}$$

Following Robert (1995), the simulation of truncated normal variables are performed using a accept-reject algorithm. Let us denote by $\mathcal{N}_+(\mu,\mu^-,\sigma^2)$ the truncated normal distribution with left truncation point μ^-. Then, at iteration k, we use the following algorithm to generate a random variable from $\mathcal{N}_+(\mu,0,1)$ where $\mu = X_{ij}\beta_k + Z_{ij}\eta_i^{(k+1)}$ and $sign(\omega_{ij}) = y_{ij} = 1$:

1. Generate $c \sim \text{Exp}(\alpha,0)$ with $\alpha = \frac{\sqrt{\mu^2+4}}{2}$;
2. Compute $\rho(c) = \exp[-(z-\alpha)^2/2]$;
3. Generate $u \sim \mathcal{U}(0,1)$ and take $\omega_{ij}^{(k+1)} = c$ if $u \leq \rho(c)$; otherwise, repeat from Step 1.

If $\omega \sim \mathcal{N}_-(\mu,\mu^+,1)$, the right–truncated normal distribution at truncated point $\mu^+ = 0$, we have that $-\omega \sim \mathcal{N}_+(\mu, -\mu^+,1)$, then the simulation from the right–truncated normal distribution is directly derived.

Estimation of the likelihood

The likelihood of the observation can be decomposed as follows

$$\begin{aligned} g(\boldsymbol{y};\boldsymbol{\theta}) &= \int p(\boldsymbol{y},\boldsymbol{\eta};\boldsymbol{\theta})d\boldsymbol{\eta} \\ &= \int h(\boldsymbol{y}|\boldsymbol{\eta};\boldsymbol{\theta})\pi(\boldsymbol{\eta};\boldsymbol{\theta})d\boldsymbol{\eta} \end{aligned}$$

where π is the prior distribution of $\boldsymbol{\eta}$. According to section 5.3, π is a Gaussian distribution and $\boldsymbol{y_i}|\boldsymbol{\eta};\boldsymbol{\theta} \sim Ber(\Phi(\boldsymbol{X_i\beta} + \boldsymbol{Z_i\eta_i}))$.

For any distribution $\tilde{\pi}$ absolutely continuous with respect to the prior distribution π, we can write

$$g(\boldsymbol{y};\boldsymbol{\theta}) = \int h(\boldsymbol{y}|\boldsymbol{\eta};\boldsymbol{\theta})\frac{\pi(\boldsymbol{\eta};\boldsymbol{\theta})}{\tilde{\pi}(\boldsymbol{\eta};\boldsymbol{\theta})}\tilde{\pi}(\boldsymbol{\eta};\boldsymbol{\theta})d\boldsymbol{\eta}.$$

We approximate $g(\boldsymbol{y};\boldsymbol{\theta})$ via an Importance Sampling integration method:

1. draw $\boldsymbol{\eta}^{(1)},\boldsymbol{\eta}^{(2)},\cdots,\boldsymbol{\eta}^{(M)}$ with the distribution $\tilde{\pi}(\cdot;\boldsymbol{\theta})$,

2. let

$$l_M(\boldsymbol{y};\boldsymbol{\theta}) = \frac{1}{M}\sum_{i=1}^{M} h(\boldsymbol{y}|\boldsymbol{\eta}^{(j)};\boldsymbol{\theta})\frac{\pi(\boldsymbol{\eta}^{(j)};\boldsymbol{\theta})}{\tilde{\pi}(\boldsymbol{\eta}^{(j)};\boldsymbol{\theta})}. \tag{5.4.9}$$

We estimate empirically the conditional mean $\mathbb{E}(\boldsymbol{\eta}|\boldsymbol{\omega},\boldsymbol{y};\boldsymbol{\theta})$ and the conditional variance $\text{Var}(\boldsymbol{\eta}|\boldsymbol{\omega},\boldsymbol{y};\boldsymbol{\theta})$ of $\boldsymbol{\eta}$ using the simulated sample $(\boldsymbol{\eta}^{(K_b+1)},\boldsymbol{\eta}^{(K_b+2)},\cdots,\boldsymbol{\eta}^{(K)})$ obtained in the SAEM algorithm for estimating $\boldsymbol{\theta}$ and after K_b burning iterations. Then, the sampling distribution $\tilde{\pi}$ is defined as the Gaussian distribution with these estimated parameters.

Standard error approximation

It is possible to obtain an estimation of the Fisher information matrix using the Louis's missing information principle (Louis, 1982). The matrix $\partial^2_{\boldsymbol{\theta}} \log l(\hat{\boldsymbol{\theta}})$ can be approximated by the sequence (H_k) defined as follows:

$$\begin{aligned} \Delta_k &= \Delta_{k-1} + \delta_k[\partial_{\boldsymbol{\theta}} \log f(\boldsymbol{y},\boldsymbol{z}^{(k)};\boldsymbol{\theta}_k) - \Delta_{k-1}] \\ G_k &= G_{k-1} + \delta_k(\partial^2_{\boldsymbol{\theta}} \log f(\boldsymbol{y},\boldsymbol{z}^{(k)};\boldsymbol{\theta}_k) + \partial_{\boldsymbol{\theta}} f(\boldsymbol{y},\boldsymbol{z}^{(k)};\boldsymbol{\theta}_k)\partial_{\boldsymbol{\theta}} f(\boldsymbol{y},\boldsymbol{z}^{(k)};\boldsymbol{\theta}_k)' - G_{k-1}) \\ H_k &= G_k - \Delta_k\Delta_k'. \end{aligned} \tag{5.4.10}$$

The sequence of stepsizes (δ_k) is as follows

$$\delta_k = \begin{cases} 1 & \text{for } 1 \leq k \leq K_b \\ \frac{1}{k-K_b} & \text{for } K_b \leq k \leq K \end{cases}, \tag{5.4.11}$$

where K_b is the number of burning iterations. Lavielle (2005) recommend to choose $K_b > K_1$, with K_1 the number of iterations with the stepsizes $\gamma_k = 1$.

5.5 The PX–SAEM algorithm for binary data

As we discussed in Chapter 3, the SAEM algorithm can be slow to converge in some situations. The PX–SAEM algorithm allows to speed up the standard SAEM algorithm introducing a working parameter. The PX-SAEM algorithm expands the complete-data model parametrized by $\boldsymbol{\theta}$, to a larger model parametrized by $\boldsymbol{\Theta}$, with $\boldsymbol{\Theta} = (\boldsymbol{\theta},\alpha)$ and where α is a working parameter. There exists a many–to–one reduction function $R : \boldsymbol{\Theta} \rightarrow R(\boldsymbol{\theta})$ which preserves the original observed-data model, and a value α_0 of α that preserves the original complete-data model.
In this section, we adapt this PX version of SAEM in the context of GLMM to improve converge of SAEM. Two versions of PX–SAEM are proposed, introducing different working parameters and obtaining two different models.

5.5.1 A first version of PX–SAEM

Let the model defined in (5.3.4) and let the working parameter be the scalar σ. We introduce this working parameter as follows: $\omega^\star_{ij} = \sigma\omega_{ij}$. The expand model is then:

$$\begin{aligned} y_{ij} &= sign(\omega^\star_{ij}) \\ \omega^\star_{ij} &= X_{ij}\boldsymbol{\beta}\sigma + Z_{ij}\boldsymbol{\eta_i}\sigma + \varepsilon_{ij}\sigma \end{aligned} \tag{5.5.12}$$

where $\varepsilon_{ij} \sim \mathcal{N}(0,1)$. Then if we define $\boldsymbol{\eta_i}^\star = \sigma\boldsymbol{\eta_i}$, $\boldsymbol{\beta}^\star = \sigma\boldsymbol{\beta}$, $\boldsymbol{\Gamma}^\star = \sigma^2\boldsymbol{\Gamma}$ and $\boldsymbol{\Theta} = (\sigma,\boldsymbol{\beta}^\star,\boldsymbol{\Gamma}^\star)$ we have:

Model $\mathcal{M}_1$

$$y_{ij}|\boldsymbol{\eta_i};\Theta \sim Ber\left(\Phi\left(\frac{X_{ij}\boldsymbol{\beta}^\star + Z_{ij}\boldsymbol{\eta_i}^\star}{\sigma}\right)\right) \tag{5.5.13}$$

$$\boldsymbol{\omega_i}^\star \sim \mathcal{N}(\boldsymbol{X_i\beta}^\star + \boldsymbol{Z_i\eta_i}^\star, \sigma^2 \times I_{n_i}). \tag{5.5.14}$$

To use the PX–SAEM algorithm, two conditions must be satisfied: there exists a many–to–one reduction function R which preserves the original observed–data model and a value σ_0 of σ that preserves the original complete–data model. In this context, $R(\boldsymbol{\Theta}) = (\boldsymbol{\beta}^\star/\sigma,\boldsymbol{\Gamma}^\star/\sigma^2)$ and $\sigma_0 = 1$.

The complete data log likelihood is:

$$\begin{aligned} l(\boldsymbol{\Theta}) &= -\frac{N_{tot}}{2}\log 2\pi - \frac{N_{tot}}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{n_i}(\omega_{ij}^\star - X_{ij}\boldsymbol{\beta}^\star - Z_{ij}\boldsymbol{\eta_i}^\star)^2 \\ &- \frac{N}{2}\log 2\pi - \frac{N}{2}\log|\boldsymbol{\Gamma}^\star| - \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{\eta_i}^{\star\prime}\boldsymbol{\Gamma}^{\star^{-1}}\boldsymbol{\eta_i}^\star, \end{aligned}$$

where $N_{tot} = \sum_{i=1}^{N} n_i$.

It is easy to deduce that the minimal sufficient statistics for $\boldsymbol{\beta}^\star$, $\boldsymbol{\Gamma}^\star$ and σ are:

$$S_1^\star = \sum_{i=1}^{N}\boldsymbol{X_i}'(\boldsymbol{\omega_i}^\star - \boldsymbol{Z_i\eta_i}^\star) \tag{5.5.15}$$

$$S_2^\star = \sum_{i=1}^{N}\boldsymbol{\eta_i}^\star\boldsymbol{\eta_i}^{\star\prime} \tag{5.5.16}$$

$$S_3^\star = \sum_{i=1}^{N}\sum_{j=1}^{n_i}\left(\omega_{ij}^\star - X_{ij}\boldsymbol{\beta}^\star - Z_{ij}\eta^\star\right)^2, \tag{5.5.17}$$

respectively.

Because the observed–data model does not depend on σ^2, we can set σ^2 to σ_0^2 at the beginning of each Simulation Step. Then, at iteration k, the PX–SAEM is as follows:

- *PX1–Simulation Step*: unchanged step because $\sigma^2 = \sigma_0^2$ then $\omega_{ij}^\star = \omega_{ij}$ and $\boldsymbol{\eta_i}^\star = \boldsymbol{\eta_i}$,

- *PX1–Stochastic Approximation Step*: update the sufficient statistics of the complete model:

$$\begin{aligned} s_{1,k+1}^\star &= s_{1,k}^\star + \gamma_k \left(S_{1,k+1}^\star - s_{1,k}^\star\right) \\ s_{2,k+1}^\star &= s_{2,k}^\star + \gamma_k \left(S_{2,k+1}^\star - s_{2,k}^\star\right) \\ s_{3,k+1}^\star &= s_{3,k}^\star + \gamma_k \left(S_{3,k+1}^\star - s_{3,k}^\star\right). \end{aligned}$$

- *PX1–Maximization Step*: compute $\widehat{\Theta}_{k+1}$ that maximizes $Q_{k+1}(\Theta)$ and apply the reduction function to obtain $\theta_{k+1} = R(\widehat{\Theta}_{k+1})$ and $\Theta_{k+1} = (\theta_{k+1}, \sigma_0^2)$.

The PX1–Maximization step updates the parameters as follows:

$$\boldsymbol{\beta_{k+1}}^\star = \left(\sum_{i=1}^{N}\sum_{j=1}^{n_i} X_{ij}' X_{ij}\right)^{-1} s_{1,k+1}^\star \tag{5.5.18}$$

$$\Gamma_{k+1}^\star = \frac{s_{2,k+1}^\star}{N} \tag{5.5.19}$$

$$\sigma_{k+1}^2 = \frac{s_{3,k+1}^\star}{N_{tot}}. \tag{5.5.20}$$

5.5.2 A second version of PX–SAEM

Let us define the following one to one linear transformation $\boldsymbol{\eta_i} = T\boldsymbol{\eta_i}^\star$ where $T = \{t_{ij}\}$ for $i,j = 1, \cdots, d$. In the "scaled" version of the EM algorithm (Foulley and Quaas, 1995; Meng and van Dyk, 1998), T is a Cholesky transformation such that $\boldsymbol{\eta_i}^\star \sim \mathcal{N}(0, I_d)$. Here, T is a full matrix of coefficients used as the auxiliary parameter $\text{vec}(T) = (t_{11}, t_{21}, t_{22}, \cdots, t_{dd})$. The latent variable model now can be written as

$$\boldsymbol{\omega} = \boldsymbol{X\beta} + \boldsymbol{Z_1\eta_1} + \cdots + \boldsymbol{Z_d\eta_d} + \boldsymbol{\varepsilon}$$

i.e. after transformation defined previously

$$\boldsymbol{\omega} = \boldsymbol{X\beta} + t_{11}\boldsymbol{Z_1\eta_1}^\star + \cdots + t_{dd}\boldsymbol{Z_d\eta_d}^\star + \boldsymbol{\varepsilon}$$

where $\boldsymbol{X\beta}$ is as before and $\boldsymbol{Z_1}, \cdots, \boldsymbol{Z_d}$ are the incidence matrices pertaining to $\boldsymbol{\eta_1}, \cdots, \boldsymbol{\eta_d}$ respectively.

Notice that the model has a linear structure in β and vec(T) provided the η_k^*'s are observed i.e. $\omega = W\alpha$ with $W = (X, Z_1\eta_1^*, \cdots, Z_d\eta_d^*)$ and $\alpha = (\beta', \text{vec}(T))'$.

What we will do is the following:

i) compute Γ^* as in a standard SAEM using the property that η and η^* have the same distribution for T equal to its reference value $T_0 = I_{d\times d}$.

ii) Then $\alpha = (\beta', \text{vec}(T))'$ can be updated using the linear structure $\omega = W\alpha$ and its corresponding sufficient complete data statistics $W'\omega$.

Then the PX–SAEM under model $\mathcal{M}_2$ is as follows:

- *PX2–Simulation Step*: unchanged step because $T = I$ then η_i and η_i^* have the same distribution.

- *PX2–Stochastic Approximation Step*: update the sufficient statistics of the complete model:

$$s_{1,k+1}^* = s_{1,k}^* + \gamma_k \left(\left(\sum_{ij} W_{ij}^{(k+1)\prime} W_{ij}^{(k+1)} \right)^{-1} \sum_{ij} W_{ij}^{(k+1)\prime} \omega_{ij} - s_{1,k}^* \right)$$

$$s_{2,k+1}^* = s_{2,k}^* + \gamma_k \left(\sum_i \eta_i^{*(k+1)} \eta_i^{*(k+1)\prime} - s_{2,k}^* \right).$$

- *PX2–Maximization Step*: compute $\widehat{\Theta}_{k+1}$ that maximizes $Q_{k+1}(\Theta)$ and apply the reduction function to obtain $\theta_{k+1} = R(\widehat{\Theta}_{k+1}) = (\beta^*, T\Gamma^*T')$ and $\Theta_{k+1} = (\theta_{k+1}, T = T_0)$.

The PX2–Maximization step updates the parameters as follows:

$$\begin{pmatrix} \beta_{k+1}^* \\ vec(T_{k+1}) \end{pmatrix} = s_{1,k+1}^* \tag{5.5.21}$$

$$\Gamma_{k+1}^* = s_{2,k+1}^*. \tag{5.5.22}$$

Remark: For both versions introduced here, the PX–SAEM algorithm is useful only during the first iterations of the algorithm. Therefore we recommend to use $\gamma_k = 1$

during these first iterations until the sequence (θ_k) reaches this neighborhood and randomly fluctuates around the maximum. When $\gamma_k = 1$, $\widehat{\Theta}_{k+1}$ maximizes the complete-data log-likelihood $\log f(y,\eta^{(k+1)};\theta)$. The almost sure convergence of the algorithm to this maximum is then ensured by using a decreasing sequence (γ_k) without any parameter expansion.

It was shown by Delyon et al. (1999) and Kuhn and Lavielle (2004) that SAEM converges to a maximum (local or global) of the likelihood of the observations under very general conditions. Convergence of PX-SAEM is ensured under the same conditions since the parameter expansion is introduced only during the first iterations of the algorithm.

5.6 REML Estimation via SAEM

In the framework of GLMM, the approximate likelihood techniques such as the penalized quasi–likelihood (PQL, Breslow and Clayton, 1993), are known to produce severely biased estimates of both regression parameters and variance components, particularly when the response is binary and/or the variance components are large (Breslow and Lin, 1995; Lin and Breslow, 1996; Neuhaus and Segal, 1997). The Restricted Maximum Likelihood (REML) estimation procedure permits, in linear and nonlinear mixed effects models, to reduce the bias observed with the Maximum Likelihood (ML) estimation for variance components. A first possibility to apply REML in the framework of GLMM is proposed by Liao and Lipsitz (2002). They proposed to correct the bias in the profile score function of the variance components but this algorithm proved to be extremely time consuming. It requires to integrate out the random effects, use simulation to estimate the bias and then adjust for the bias. As an alternative, we propose to use SAEM to obtain the REML estimation for variance component in this context. We use the Harville's REML interpretation (see Harville, 1974; and Chapter 4) integrating out the fixed effects of the model.

Following Foulley and Quaas (1995), we consider the fixed effects as random, assuming that the prior distribution of β is noninformative. The vector of parameters θ becomes $\theta^\star = (\Gamma)$, and the vector of random effects now includes β and will be noted $z_1 = (\omega,\eta,\beta)$.

To perform REML, we use the Henderson's mixed model equations (ignoring subscripts), i.e.,

$$\begin{bmatrix} X'X & X'Z \\ Z'X & Z'Z+\Gamma^{-1} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \tilde{\eta} \end{bmatrix} = \begin{bmatrix} X'\omega \\ Z'\omega \end{bmatrix}, \tag{5.6.23}$$

where $\hat{\beta}$ is the Generalized Least Square (GLS) estimate of β and $\tilde{\eta}$ is $\mathbb{E}(\eta|\omega^{(k)},\beta^{(k)},y;\theta_k^\star)$.

Let $\boldsymbol{A} = \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \tilde{\boldsymbol{\eta}} \end{bmatrix}$ and $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{X}'\boldsymbol{X} & \boldsymbol{X}'\boldsymbol{Z} \\ \boldsymbol{Z}'\boldsymbol{X} & \boldsymbol{Z}'\boldsymbol{Z}+\boldsymbol{\Gamma}^{-1} \end{bmatrix}^{-1}$ the block of the inverse of the coefficient matrix (5.6.23). Following (5.4.5), the minimal sufficient statistic for $\boldsymbol{\Gamma}$ is $\tilde{S} = \sum_{i=1}^{N} \boldsymbol{\eta}_i \boldsymbol{\eta}_i'$.

Then, at iteration k, the SAEM–REML algorithm is therefore composed of the following steps:

- *Simulation Step:* a Gibbs scheme is used to perform the simulation step:

  1) $(\boldsymbol{\eta_i}^{(k+1)}, \boldsymbol{\beta}^{(k+1)}) | \boldsymbol{\omega}^{(k)}, \boldsymbol{y}; \boldsymbol{\theta}_k^{\star} \sim \mathcal{N}(\boldsymbol{A}, \boldsymbol{C})$

  2) $\omega_{ij}^{(k+1)} | \boldsymbol{\eta}^{(k+1)}, \boldsymbol{\beta}^{(k)}, \boldsymbol{y}; \boldsymbol{\theta}_k^{\star} \sim \mathcal{N}_{\pm}\left(X_{ij}\boldsymbol{\beta_k} + Z_{ij}\boldsymbol{\eta_i}^{(k+1)}, 1\right)$
  the truncated normal distribution in 0
  so that $sign(\omega_{ij}) = y_{ij}$.

- *Stochastic Approximation Step:* the sufficient statistical of the model is update as follows

$$s_{k+1} = s_k + \gamma_k \left(\tilde{S}_{k+1} - s_k\right).$$

- *Maximization step:* update $\theta_k^{\star}$:

$$\Gamma_{k+1} = \frac{s_{k+1}}{N}.$$

In fact, step 1 can be split itself into two substeps based on the two conditional distributions $\boldsymbol{\eta_i}|\boldsymbol{\beta},\boldsymbol{\omega},\boldsymbol{y};\boldsymbol{\theta}$ and $\boldsymbol{\beta}|\boldsymbol{\eta},\boldsymbol{\omega},\boldsymbol{y};\boldsymbol{\theta}$. This avoids the burden of inversion involved in $\boldsymbol{C}$ for a large coefficient matrix.

5.7 Applications

In this section, we illustrate the SAEM algorithm in the context of GLMM with two examples. The first application studied here allows also to show the performance of the PX version of SAEM using the two strategies proposed in this chapter, obtained a significant converge speed improvement for the parameter estimation but also for the estimated log–likelihood of the observation. The second example permits to illustrate the properties of the REML estimates compared to ML in GLMM.

5.7.1 Example 1: Epileptics data

We studied the data set composed by a clinical trial of 59 epileptics, reported by Thall and Vail (1990). For each patient, the number of epileptic seizures was recorded during a baseline period of eight weeks. Patients were then randomized to treatment with the anti-epileptic drug progabide, or to a placebo in addition to standard chemotherapy. The number of seizures was then recorded in four consecutive two-weeks intervals.

Let p_{ij} the probability that individual i has 5 (or more) seizures in the two weeks period prior visit j. We consider the following probit mixed model:

$$\begin{aligned} y_{ij} &\sim \text{Ber}(p_{ij}), \quad \text{with} \\ \Phi^{-1}(p_{ij}) &= \beta_1 + \beta_2 * treat + \beta_3 * age + \beta_4 * base + \eta_i \quad (\mathcal{P}_1) \\ & 1 \leq i \leq N = 59, \quad 1 \leq j \leq n_i = n = 4. \end{aligned}$$

Then, if we consider a latent variable, as defined in (5.3.4), we have:

$$\omega_i = \boldsymbol{X}_i\beta + \boldsymbol{Z}_i\eta_i + \varepsilon_i$$

where the matrices X_i and Z_i are as follows:

$$X_i = \begin{pmatrix} 1 & treat_{i,1} & age_{i,1} & base_{i,1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & treat_{i,n} & age_{i,n} & base_{i,n} \end{pmatrix}_{4\times 4} \quad \text{and} \quad Z_i = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{4\times 1}.$$

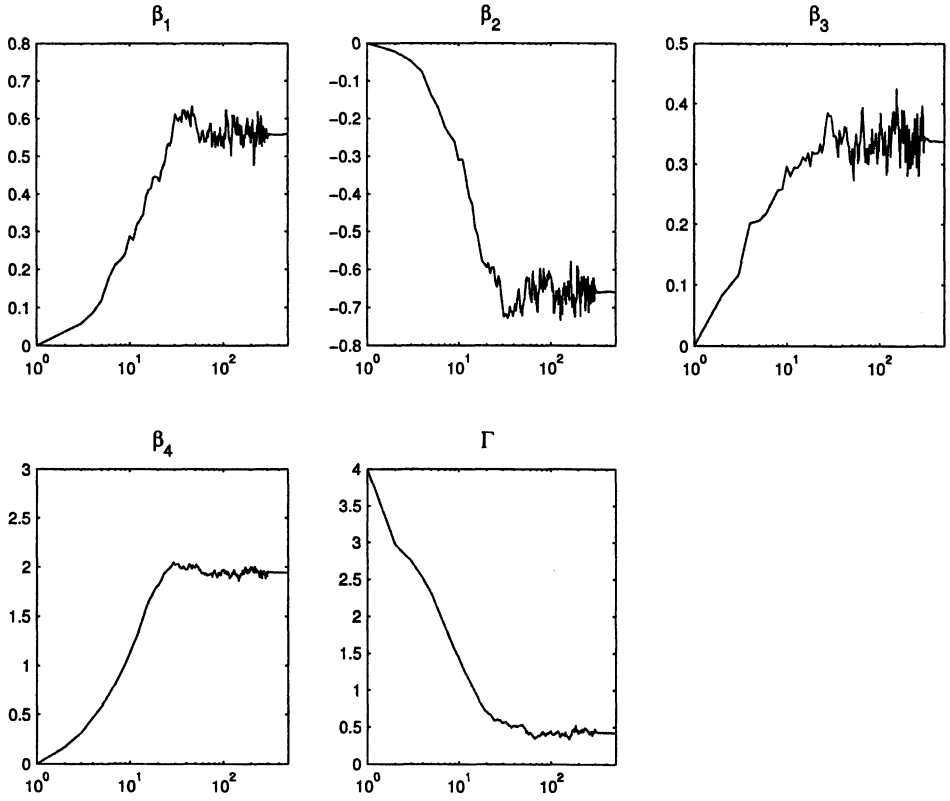
To obtain the MLE, we used the SAEM algorithm with 500 iterations, where γ_k, used in the stochastic approximation step, is equal to 1 in the 300 first iterations. The starting values of the parameters are chosen at the following values:

$$\beta^{(0)} = (0,0,0,0), \quad \Gamma^{(0)} = 4.$$

Table 5.1 and Figure 5.1 resume the results obtained with SAEM, using the ML method. In Table 5.1, are displayed the results obtained with our algorithm and the SAS NLMIXED PROC with adaptive quadrature, showing that estimates of parameters obtained with this two different algorithms are very close to each other.

We also applied the parameter expansion version of SAEM to this data set. To compare the perform of SAEM against PX–SAEM, we used different initial values. In addition, we used SAS with the same initialization. The expanded model is used in both version of

FIG. 5.1 – *Epileptics data: Estimation of θ using SAEM. A logarithmic scale is used for the x–axis.*



TAB. 5.1 – *Epileptics data: Estimation with SAEM and SAS (ML). In the first column, the initial values are enclosed in parentheses.*

	SAEM	SAS
Parameters	Estimation (S.E.)	Estimation (S.E.)
$\beta_1(0)$	0.56 (0.22)	0.56 (0.24)
$\beta_2(0)$	-0.66 (0.29)	-0.65 (0.31)
$\beta_3(0)$	0.34 (0.23)	0.36 (0.25)
$\beta_4(0)$	1.94 (0.23)	1.94 (0.30)
$\Gamma(4)$	0.43 (0.29)	0.43 (0.28)
-2L	197.4	197.4

PX–SAEM during the first 40 iterations and the standard SAEM with the original model is used after these 40 iterations.

We initialized the algorithms at different values, first not far to the MLE obtained with SAS (see results in Table 5.1), and then we chosen initial values more distant. The results are resumed in Table 5.2 and Figures 5.2 and 5.3. In Figures 5.4 and 5.5, we show the evolution of the observed log–likelihood estimated at each iteration.

Clearly, in the second case, when we initialize the algorithms far off, we see that PX–SAEM converges much faster to the MLE than SAEM. The standard SAEM algorithm takes at least two times longer to reach of the MLE, for most parameters. Figure 5.5 shows that the maximum of the observed log–likelihood is reached in few iteration with both version of PX–SAEM, but the first version of PX–SAEM seems more efficient than the second one.

Furthermore, it is important to note that SAS does not convergence when initial values are distant counter to the SAEM and the PX–SAEM algorithms. The SAEM algorithm is more robust with regard to the choice of the starting values than NLMIXED.

TAB. 5.2 – *Epileptics data: Estimation with SAEM, PX-SAEM and SAS. In the first column, the initial values are enclosed in parentheses.*

After 500 iterations

	Estimation (S.E.)			
Parameters	SAEM	PX-SAEM1	PX-SAEM2	SAS
$\beta_1(2)$	0.56 (0.20)	0.57 (0.20)	0.54 (0.21)	0.56 (0.24)
$\beta_2(2)$	-0.67 (0.29)	-0.65 (0.29)	-0.63 (0.30)	-0.65 (0.31)
$\beta_3(2)$	0.33 (0.24)	0.34 (0.21)	0.31 (0.22)	0.36 (0.25)
$\beta_4(2)$	1.95 (0.37)	1.93 (0.24)	1.93 (0.24)	1.94 (0.30)
$\Gamma(4)$	0.42 (0.47)	0.44 (0.21)	0.44 (0.21)	0.43 (0.28)
-2L	197.4	197.4	197.4	197.4

After 700 iterations

	Estimation (S.E.)			
Parameters	SAEM	PX-SAEM1	PX-SAEM2	SAS*
$\beta_1(5)$	0.56 (0.20)	0.54 (0.20)	0.57 (0.22)	
$\beta_2(5)$	-0.65 (0.26)	-0.64 (0.29)	-0.64 (0.28)	
$\beta_3(5)$	0.33 (0.22)	0.34 (0.22)	0.35 (0.23)	
$\beta_4(5)$	1.94 (0.22)	1.91 (0.24)	1.95 (0.21)	
$\Gamma(4)$	0.40 (0.24)	0.43 (0.37)	0.44 (0.27)	
-2L	197.4	197.4	197.4	

$\star$: convergence not reached

FIG. 5.2 – *Epileptics data: Estimation of θ using SAEM and PX-SAEM with $\beta_1^{(0)} = \beta_2^{(0)} = \beta_3^{(0)} = \beta_4^{(0)} = 2$. A logarithmic scale is used for the x-axis.*

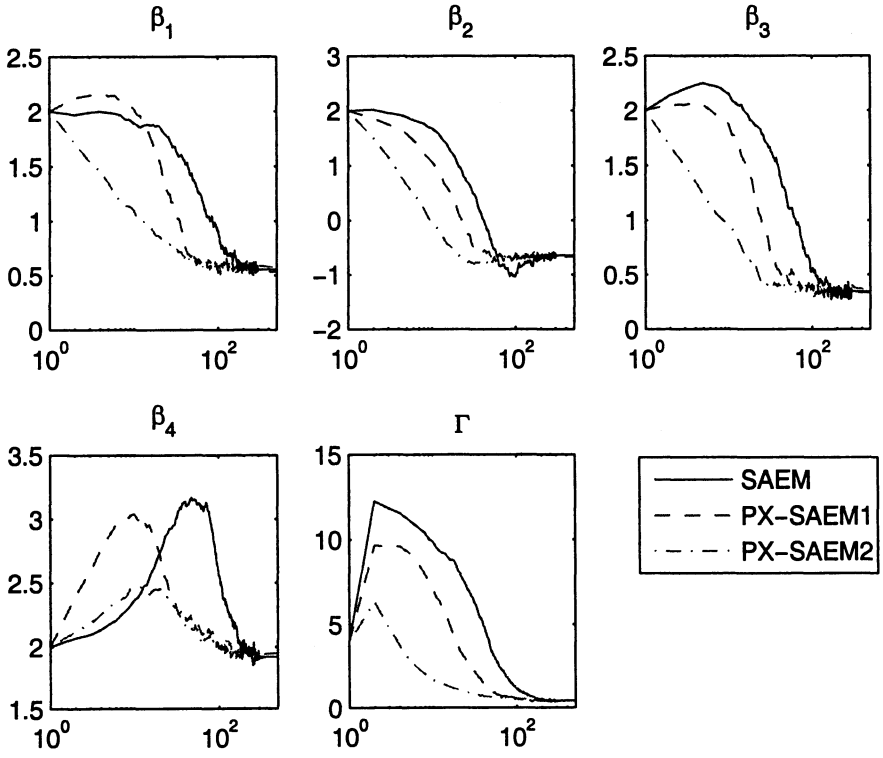


FIG. 5.3 – *Epileptics data: Estimation of θ using SAEM and PX-SAEM with $\beta_1^{(0)} = \beta_2^{(0)} = \beta_3^{(0)} = \beta_4^{(0)} = 5$. A logarithmic scale is used for the x-axis.*

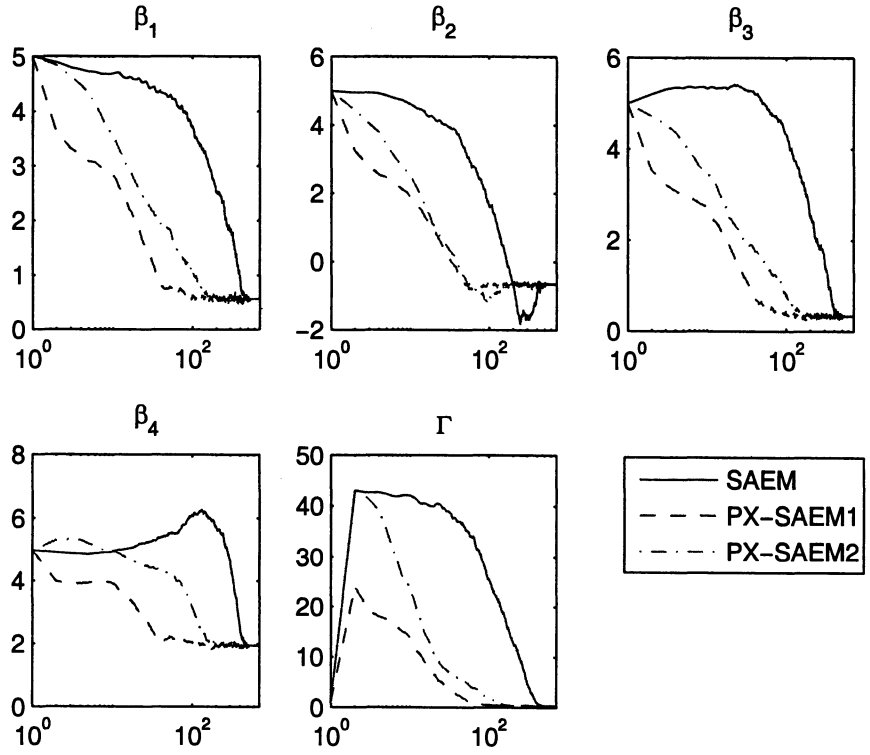


FIG. 5.4 – *Epileptics data: Estimation of the observed log-likelihood using SAEM and PX-SAEM with* $\beta_1^{(0)} = \beta_2^{(0)} = \beta_3^{(0)} = \beta_4^{(0)} = 2$. *A logarithmic scale is used for the x-axis.*

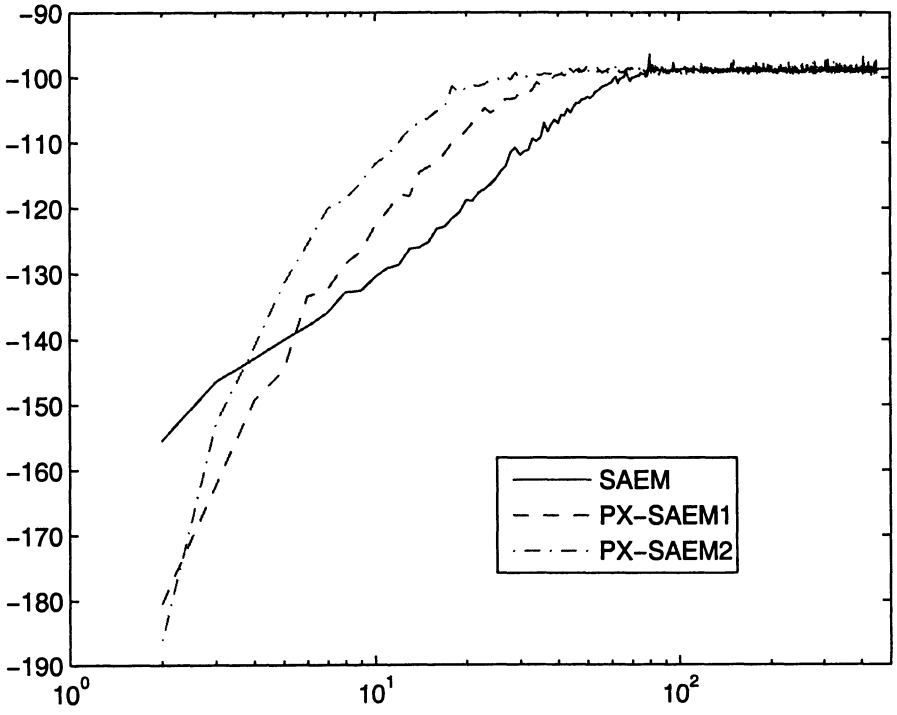
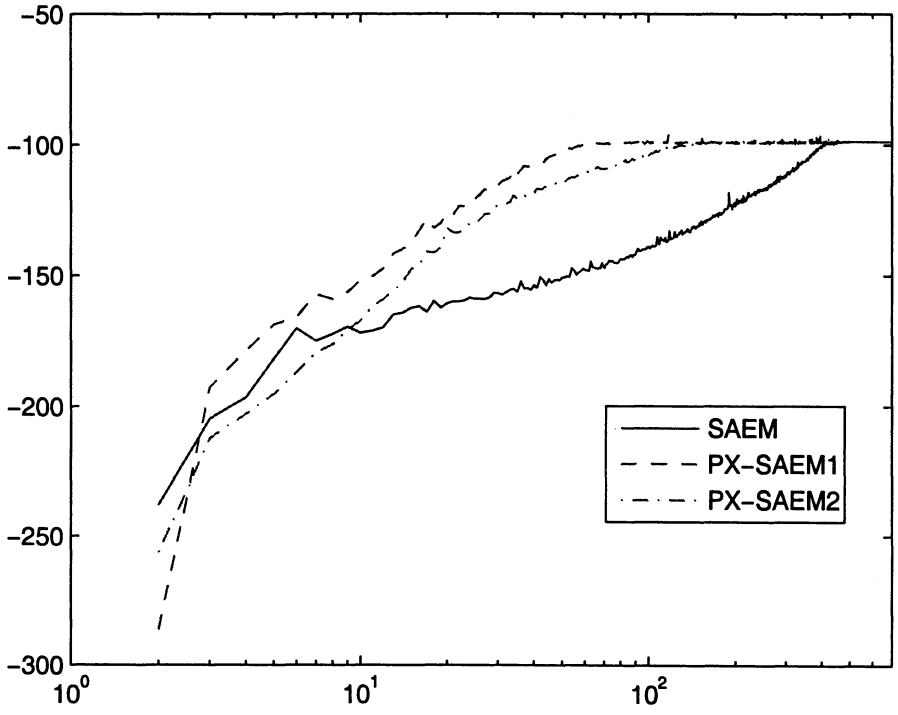


FIG. 5.5 – *Epileptics data: Estimation of the observed log-likelihood using SAEM and PX-SAEM with* $\beta_1^{(0)} = \beta_2^{(0)} = \beta_3^{(0)} = \beta_4^{(0)} = 5$. *A logarithmic scale is used for the x-axis.*



5.7.2 Example 2: Schizophrenia study

In this second example, we examined a longitudinal data, studied previously by Gibbons and Hedecker (1994), collected in the National Institute of Mental Health Schizophrenia Collaborative Study on treatment related changes in overall severity . We focused on item 79 ("Severity of Illness") of the Inpatient Multidimensional Psychiatric Scale (IMPS). This item was originally scored on a 7–point scale ranging but we dichotomized as follows:

1	=	normal	0
2	=	bordeline mentally ill	
3	=	mildly ill	
4	=	moderately ill	1
5	=	markedly ill	
6	=	severely ill	
7	=	among the most extremely ill	

This observed binary response variable is then the indicator $y_{ij} = I\{\omega_{ij} > 0\}$, for the subject i at the time T_{ij} of treatment, for the treatment group G_{ij}, $i = 1,\cdots,N$, $j = 1,\cdots,n_i$. Fixed effects included the dummy-coded drug effect (placebo = 0 and drug = 1), the time effect (square root of week) and the drug by time interaction. The underlying linear mixed model is defined as follows:

$$\omega_{ij} = \begin{pmatrix} 1 & G_{ij} & T_{ij} & G_{ij} * T_{ij} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} + \begin{pmatrix} 1 & T_{ij} \end{pmatrix} \begin{pmatrix} \eta_{1i} \\ \eta_{2i} \end{pmatrix} + \varepsilon_{ij} \qquad (5.7.24)$$

where $\boldsymbol{\eta_i} = \begin{pmatrix} \eta_{1i} \\ \eta_{2i} \end{pmatrix} \sim \mathcal{N}(0,\Gamma)$ and $\varepsilon_i \sim \mathcal{N}(0,1)$.

There are large differences in the number of measurement made in the 6 weeks of treatment. Table 5.3 shows the experimental design and corresponding samples sizes.

In this example, we used the SAEM algorithm with 800 iterations, where γ_k, used in the stochastic approximation step, is equal to 1 in the 500 first iterations. The starting values of the parameters are chosen at the following values:

$$\boldsymbol{\beta}^{(0)} = \left(\beta_1^{(0)},\beta_2^{(0)},\beta_3^{(0)},\beta_4^{(0)}\right)^T = (0,0,0,0)^T \quad \text{and} \quad \Gamma^{(0)} = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}.$$

TAB. 5.3 – *"Schizophrenia study": Experimental design and samples sizes*

	Sample size at Week						
Group	0	1	2	3	4	5	6
Placebo ($N = 108$)	107	105	5	87	2	2	70
Drug ($N = 329$)	327	321	9	287	9	7	265

TAB. 5.4 – *"Schizophrenia study": ML Estimates (SAEM and SAS) and REML Estimates (SAEM–REML). The random effects correlation is noted ρ.*

	ML		REML
Parameters	SAEM	SAS	SAEM–REML
$\beta_1(0)$	3.291	3.220	–
$\beta_2(0)$	0.211	0.198	–
$\beta_3(0)$	-0.763	-0.734	–
$\beta_4(0)$	-0.936	-0.922	–
$\Gamma_1(4)$	2.16	2.016	2.66
$\Gamma_2(4)$	1.03	0.984	1.13
$\rho(0)$	-0.47	-0.46	-0.48

FIG. 5.6 – *"Schizophrenia Study": Maximum Likelihood Estimates of the vector of parameters θ using SAEM. A logarithmic scale is used for the x-axis.*

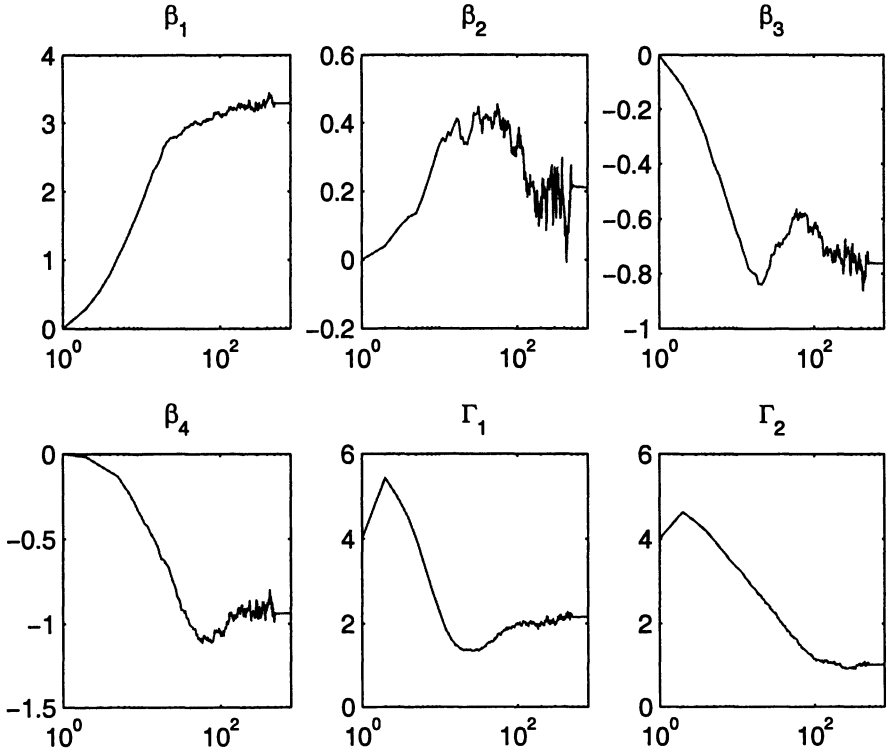
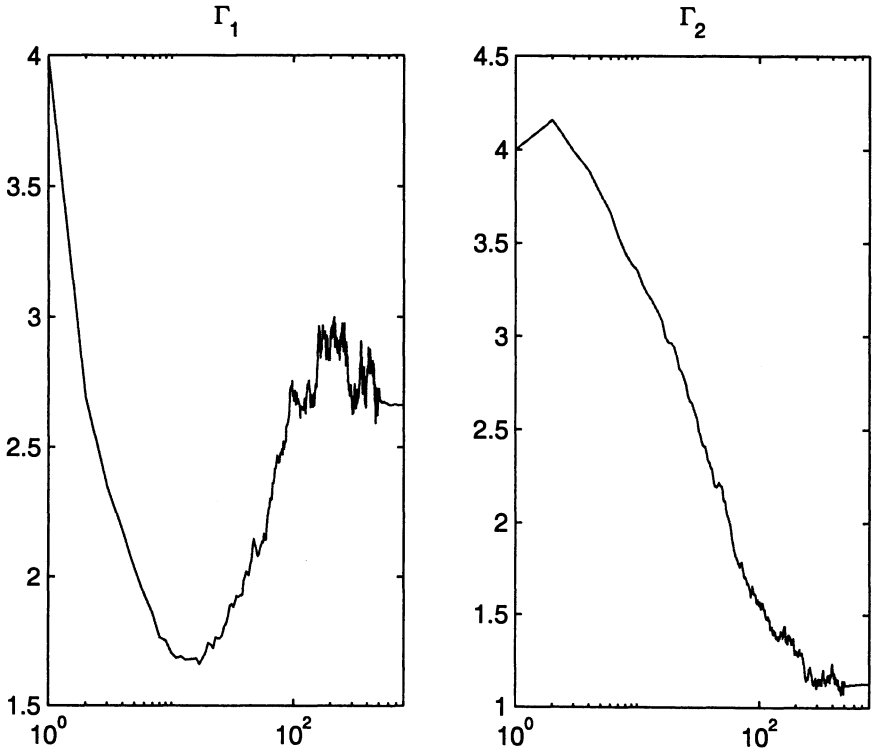


FIG. 5.7 – *"Schizophrenia Study": Restricted Maximum Likelihood Estimates of variance components using SAEM. A logarithmic scale is used for the x–axis.*



Results in Table 5.4 show that there are not large differences between the ML estimates obtained with SAEM and SAS. As can be expected, the REML estimations are larger than ML estimations for the variance components in particular for Γ_1. This was to be expected as the ML estimate are known, in linear and nonlinear mixed effects, to be biased downwards specially when the data are unbalanced. It seems that REML estimation procedure proposed here was more robust than ML for unbalanced design.

As a remark, we noted that NLMIXED PROC was more sensitive to the choice of the starting values than SAEM in this example. We also experienced some convergence problems with SAS.

5.8 Other research

In this section we propose several possible extensions of the SAEM algorithm in the GLMM context. A first natural extension is to consider others GLMM for binary data like the Logistic regression model. The SAEM algorithm and its extension could be an alternative to obtain ML and REML estimates. An other possible study is to consider the correlated Probit model, analized by Gueorguieva and Agresti (2001), which contemplates a joint modeling of continuous and discrete response. This model considers for each subject a single binary response and a single continuous response. It seems interesting to propose a REML methodology, using SAEM, for the estimates of the variance components in this context. Finally, it should be attractive to consider also the case of the ordinal data which is a generalization of the binary outcomes.
These problems are detailed next.

5.8.1 Others GLMM for dichotomous outcomes: the Logistic model

Many other models are used to study dichotomous data. The most popular is the logistic regression model.

For the case of binary data, as an alternative to the probit link function, it may possible to use the logit link. Then the model (5.3.3) defined in the Section 5.3, is now as follows:

$$\begin{aligned} y_{ij} &\sim \text{Ber}(p_{ij}), \quad \text{with} \\ p_{ij} &= P(y_{ij}=1) = \Psi(X_{ij}\boldsymbol{\beta} + Z_{ij}\boldsymbol{\eta_i}) \end{aligned} \tag{5.8.25}$$

$$\Rightarrow \Psi^{-1}(p_{ij}) = X_{ij}\boldsymbol{\beta} + Z_{ij}\boldsymbol{\eta_i}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq n_i$$

where $\boldsymbol{\eta_i} \in \mathbb{R}^d$ follows a Gaussian distribution $\mathcal{N}(0,\Gamma)$ and $\Psi(\cdot)$ denotes the the logistic cumulative distribution function, $\Psi = [1+\exp(-X_{ij}\boldsymbol{\beta} - Z_{ij}\boldsymbol{\eta_i})]^{-1}$. X_{ij} and Z_{ij} are two $1 \times p$ and $1 \times d$ known matrices.

In terms of the underlying latent $\boldsymbol{\omega}$, the model does not change. Then, in the application of SAEM in this kind of model, the only changes take place in the simulation step of the random effects $\boldsymbol{\omega}$ since now the conditional distribution $\boldsymbol{\omega}|\boldsymbol{\eta},\boldsymbol{y};\boldsymbol{\theta}$ is the truncated Logistic distribution.

The REML estimation and the parameter expansion version of SAEM could be also implemented easily.

5.8.2 The correlated probit model

Most research about clustered data has concentrated on a single response, but many studies have measured multiple response variables for each subject. Foulley et al. (1983) analysed dystocia in cattle using data on both birth weight and frequency of difficult calving via a multivariate response model involving binary and continuous variables. They used an Emperical Bayes approach to compute MAP estimators of fixed and random effects whereas variance components were estimated via an EM–REML type algorithm based on a gaussian approximation of the posterior distribution of random effects. More recently Gueorguieva and Agresti (2001) consider the same kind of modelling to analyse toxicity data on mice (fetal weight and malformation). They propose a Monte Carlo expectation–conditional maximization algorithm for finding MLE. Again it might be worthwhile to obtain REML estimates of variance components in this context via the SAEM algorithm.

Using the same notation of Gueorguieva and Agresti (2001), let $\{y_{i1j}\}$ denote the observed continuous measurement and $\{y^{\star}_{i2j}\}$ denote the latent continuous measurement underlying the binary response at the j^{th} subject, $i = 1,\cdots,N$, $j = 1,\cdots,n_i$. The under-

lying mixed model is defined as follows:

$$\begin{aligned} y_{i1j} &= \boldsymbol{X}_{i1j}\boldsymbol{\beta}_1 + \boldsymbol{Z}_{i1j}\boldsymbol{\eta}_{i1} + \varepsilon_{i1j} \\ y^{\star}_{i2j} &= \boldsymbol{X}_{i2j}\boldsymbol{\beta}_2 + \boldsymbol{Z}_{i2j}\boldsymbol{\eta}_{i2} + \varepsilon_{i2j}, \end{aligned}$$

where $\boldsymbol{X}_{i1j}$, $\boldsymbol{X}_{i2j}$, $\boldsymbol{Z}_{i1j}$ and $\boldsymbol{Z}_{i2j}$ are known $1 \times p_1$, $1 \times p_2$, $1 \times q_1$ and $1 \times q_2$ vectors, and $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are unknown $p_1 \times 1$ and $p_2 \times 1$ parameter vector. The random effects and the random errors are assumed to be normally distributed:

$$\begin{aligned} \boldsymbol{\eta}_i &= \begin{pmatrix} \boldsymbol{\eta}_{i1} \\ \boldsymbol{\eta}_{i2} \end{pmatrix} \sim_{i.i.d.} \mathcal{N}(0,\boldsymbol{\Gamma}) = \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12} \\ \boldsymbol{\Gamma}_{21} & \boldsymbol{\Gamma}_{22} \end{bmatrix}\right) \\ \boldsymbol{\varepsilon}_{ij} &= \begin{pmatrix} \varepsilon_{i1j} \\ \varepsilon_{i2j} \end{pmatrix} \sim_{i.i.d.} \mathcal{N}(0,\boldsymbol{\Gamma}_{\varepsilon}) = \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{e1} & \sigma_{e12} \\ \sigma_{e21} & \sigma^2_{e2} \end{bmatrix}\right). \end{aligned}$$

To compute the maximum likelihood (ML) and but above all restricted maximum likelihood (REML) estimator of the respective unknown parameter vector $\theta = (\boldsymbol{\beta},\boldsymbol{\Gamma})$ and $\theta^{\star} = (\boldsymbol{\Gamma})$, we use the SAEM algorithm.

The SAEM algorithm: ML Method

In this context, the complete data consists of $\{\boldsymbol{y}_{ij}{}^{\star}\} = \{(y_{i1j},y^{\star}_{i2j})^T\}$, $i = 1,\cdots,n$ and $\boldsymbol{\eta}_i$. Let $\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{pmatrix}$ and $\boldsymbol{X}_{ij} = \begin{pmatrix} \boldsymbol{X}_{i1j} & 0 \\ 0 & \boldsymbol{X}_{i2j} \end{pmatrix}$. Then the complete data log–likelihood is

$$\begin{aligned} l(\boldsymbol{y},\boldsymbol{\eta};\boldsymbol{\theta}) &= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{n_i}\log|\boldsymbol{\Gamma}_{\varepsilon}| - \frac{1}{2}\sum_{i=1}^{N}\log|\boldsymbol{\Gamma}| \\ &\quad -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{n_i}(\boldsymbol{y}_{ij}{}^{\star} - \boldsymbol{X}_{ij}\boldsymbol{\beta} - \boldsymbol{Z}_{ij}\boldsymbol{\eta}_i)'\boldsymbol{\Gamma}_{\varepsilon}^{-1}(\boldsymbol{y}_{ij}{}^{\star} - \boldsymbol{X}_{ij}\boldsymbol{\beta} - \boldsymbol{Z}_{ij}\boldsymbol{\eta}_i)) \\ &\quad -\frac{1}{2}\sum_{i=1}^{N}\boldsymbol{\eta}_i'\boldsymbol{\Gamma}^{-1}\boldsymbol{\eta}_i - \frac{N_{tot}+N}{2}\log(2\pi), \end{aligned}$$

where $N_{tot} = \sum_{i=1}^{N} n_i$.

At iteration k, the SAEM in this context is as follows:

• *Simulation Step*: draw η and $y_{i2j}{}^{\star}$ from the joint distribution $y_2{}^{\star},\eta|y_1,y_2;\theta_k$, where $y_1 = \{y_{i1j}\}$, $y_2 = \{y_{i2j}\}$ and $y_2{}^{\star} = \{y_{i2j}{}^{\star}\}$. We use for this a Gibbs scheme, we simulate the non observed data according the conditional distributions $\eta^{(k+1)}|y_1,y_2,y_2{}^{\star(k)};\theta_k$ (a Normal distribution) and $y_2{}^{\star(k+1)}|\eta^{(k)},y_1,y_2;\theta_k$ (a truncated Normal distribution).

• *Approximation Step*: update the sufficient statistics of the complete model:

$$\begin{aligned} s_{1i,k+1} &= s_{1,k} + \gamma_{k+1}\left[\eta_i - s_{1,k}\right] \\ s_{2i,k+1} &= s_{2,k} + \gamma_{k+1}\left[\eta_i{}^{(k+1)\prime}\eta_i{}^{(k+1)} - s_{2,k}\right]. \end{aligned}$$

• *Maximization Step*: update θ_{k+1}:

$$\begin{aligned} \beta_{k+1} &= \left(\sum_{i=1}^{N}\sum_{j=1}^{n_i} X_{ij}{}'\Gamma_{\varepsilon}^{k^{-1}}X_{ij}\right)^{-1}\sum_{i=1}^{N} X_{ij}{}'(y_{ij}{}^{\star} - Z_{ij}s_{1i,k+1}); \\ \Gamma^{k+1} &= \frac{1}{N}\sum_{i=1}^{N} s_{2i,k+1}; \\ \Gamma_{\varepsilon}^{k+1} &= \frac{1}{N_{tot}}\sum_{i=1}^{N}\sum_{j=1}^{n_i}(y_{ij}{}^{\star} - X_{ij}\beta_{k+1})^2 + \mathrm{tr}(Z_{ij}s_{2i,k+1}Z_{ij}{}') - \\ &\quad 2(y_{ij}{}^{\star} - X_{ij}\beta_{k+1})'Z_{ij}s_{1i,k+1}. \end{aligned}$$

The SAEM algorithm: REML estimation

To obtain the REML estimates of variance components in this model, we use the Harville's REML interpretation, considering the fixed effects as random (see Chapter 4 and Section 5.6 for more details). Then, the non observed data is $z_2 = (y_2{}^{\star},\eta,\beta)$ and the unknown parameter vector is $\theta = (\Gamma)$. The idea is then to use a Gibbs scheme and the Henderson's mixed model equations (5.6.23) to perform the simulation step and to simulate z_2 from the joint distribution of $z_2|y_1,y_2;\theta_k^{\star}$.

5.8.3 The normal probit model for ordinal data

Models for ordinal response variables are important in many areas of research, since subjects are often classified or may respond on an ordinal, or graded scale (see e.g. an

overview on this topic by Agresti, 1999). It is often the case that subjects are observed nested within clusters or are repeatedly. In this kind of models, we consider c ordered categories assigning the numbers $1, \cdots, c$. We observe N correlated vectors $\boldsymbol{y_i} = \{y_{i1}, \cdots, y_{ic}\}'$ where $y_{ij} = I(\text{individual } i \text{ is in category } j)$. Further we have $\sum_{j=1}^{c} y_{ij} = 1$ and we know the proportions of observation in each category:

$$n_j = \sum_{i=1}^{N} y_{ij}, \quad j = 2, \cdots, c.$$

If we consider a latent variable ω_i, the ordinal probit mixed model is as follows:

$$y_{ij} = I(t_{j-1} < \omega_{ij} \leq t_j) \tag{5.8.26}$$

$$\omega_{ij} = X_{ij}\boldsymbol{\beta} + Z_{ij}\boldsymbol{\eta_i} + \varepsilon_{ij}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq c, \tag{5.8.27}$$

where $\boldsymbol{\eta_i} \in \mathbb{R}^d$ follows a Gaussian distribution $\mathcal{N}(0,\boldsymbol{\Gamma})$. The t_j are unknown cut–off points such that $t_0 = -\infty < t_1 < \cdots < t_c = \infty$.

X_{ij} and Z_{ij} are two $1 \times p$ and $1 \times d$ known matrices. Furthermore, we have $\varepsilon_{ij} \sim \mathcal{N}(0,1)$ and we suppose that $(\boldsymbol{\eta_i})$ and (ε_{ij}) are mutually independent. We consider the unknown parameter vector $\boldsymbol{\theta} = (\boldsymbol{\beta},\boldsymbol{\Gamma},t_1, \cdots, t_{c-1})$

Conditional on the vector of random effects $\boldsymbol{\eta}$, the $(\boldsymbol{y_i})$ are independent with

$$\mathbb{E}(\boldsymbol{y_i}|\boldsymbol{\eta}) = \Phi(\boldsymbol{X_i\beta} + \boldsymbol{Z_i\eta}), \tag{5.8.28}$$

where Φ denotes the Normal cumulative distribution function.

Then we have

$$y_{ij}|\boldsymbol{\eta};\boldsymbol{\theta} \sim Ber\,(p_{ij})\,, \quad \text{with} \tag{5.8.29}$$

$$p_{ij} = P(y_{ij} = 1|\boldsymbol{\eta}) \tag{5.8.30}$$

$$= \Phi(t_j - \boldsymbol{X_i\beta} - \boldsymbol{Z_i\eta}) - \Phi(t_{j-1} - \boldsymbol{X_i\beta} - \boldsymbol{Z_i\eta}) \quad \text{and} \tag{5.8.31}$$

$$\boldsymbol{\eta};\boldsymbol{\theta} \sim \mathcal{N}(0,\boldsymbol{\Gamma}). \tag{5.8.32}$$

Our purpose is to compute the MLE of $\boldsymbol{\theta} = (\boldsymbol{\beta},\boldsymbol{\Gamma},t_1, \cdots, t_{c-1})$.

The SAEM Algorithm

We regard the missing data as the vector of unobserved random effects $\boldsymbol{\eta}$ and we consider $\boldsymbol{y} = (y_{ij}, 1 \leq i \leq N, 1 \leq j \leq c)$ as the observed data. Then, the completed data are $(\boldsymbol{y}, \boldsymbol{\eta})$.

The complete data log likelihood is:

$$l(\boldsymbol{\theta}) \propto \sum_{i=1}^{N} \sum_{j=1}^{c} y_{ij} \log(p_{ij}) - \frac{N}{2} \log|\boldsymbol{\Gamma}| - \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{\eta_i}' \boldsymbol{\Gamma}^{-1} \boldsymbol{\eta_i}, \tag{5.8.33}$$

where $p_{ij} = \Phi(t_j - \boldsymbol{X_i}\boldsymbol{\beta} - \boldsymbol{Z_i}\boldsymbol{\eta}) - \Phi(t_{j-1} - \boldsymbol{X_i}\boldsymbol{\beta} - \boldsymbol{Z_i}\boldsymbol{\eta})$.

The minimal sufficient statistics are:

$$S_1(i) = \log(\Phi(t_j - \boldsymbol{X_i}\boldsymbol{\beta} - \boldsymbol{Z_i}\boldsymbol{\eta}) - \Phi(t_{j-1} - \boldsymbol{X_i}\boldsymbol{\beta} - \boldsymbol{Z_i}\boldsymbol{\eta})) \tag{5.8.34}$$

$$S_2(i) = \boldsymbol{\eta_i}\boldsymbol{\eta_i}'. \tag{5.8.35}$$

To perform the Simulation step of the SAEM algorithm, we use the Gibbs sampler to generate variates from the conditional density $\boldsymbol{\eta}|\boldsymbol{y}$. We augment the unobserved random effects $\boldsymbol{\eta}$ with the latent variable $\boldsymbol{\omega}$.

Then, at iteration k, we have:

- *S–A Step:* draw $\boldsymbol{\eta}$ and $\boldsymbol{\omega}$ from the joint distribution $\boldsymbol{\omega}, \boldsymbol{\eta}|\boldsymbol{y}; \boldsymbol{\theta}_k$. We use for this a Gibbs scheme:

$$\boldsymbol{\eta_i}^{(k+1)}|\boldsymbol{\omega}^{(k)}, \boldsymbol{y}; \boldsymbol{\theta}_k \sim \mathcal{N}(W_i^{(k)} Z_i'(\boldsymbol{\omega_i}^{(k)} - \boldsymbol{X_i}\boldsymbol{\beta}_k), \boldsymbol{W_i}^{(k)}),$$
$$\text{where } \boldsymbol{W_i}^{(k)} = [\boldsymbol{Z_i'}\boldsymbol{Z_i} + \boldsymbol{\Gamma_k}^{-1}]^{-1}$$

$$\omega_{ij}^{(k+1)}|\boldsymbol{\eta}^{(k+1)}, \boldsymbol{y}; \boldsymbol{\theta}_k \sim \mathcal{N}_{\pm}\left(X_{ij}\boldsymbol{\beta}_k + Z_{ij}\boldsymbol{\eta_i}^{(k+1)}, 1\right)$$
$$\text{the truncated normal distribution between } (t_{j-1}, t_j)$$
$$\text{if } y_{ij} = 1.$$

Then we have:

$$s_{1,k+1}(i) = s_{1,k}(i) + \gamma_k \left(S_{1,k+1}(i) - s_{1,k}(i)\right)$$
$$s_{2,k+1}(i) = s_{2,k}(i) + \gamma_k \left(S_{2,k+1}(i) - s_{2,k}(i)\right).$$

- *M Step:* update $\boldsymbol{\theta}_k$:

  $\boldsymbol{\beta}_{k+1}$ and t_{k+1} maximize

  $$\sum_{i=1}^{N}\sum_{j=1}^{c} y_{ij} s_{1,k+1}(i)$$

  subject to the constraint $t_1 < \cdots < t_{c-1}$, and

  $\boldsymbol{\Gamma}_{k+1} = \sum_{i=1}^{N} s_{2,k+1}(i)$.

The REML estimation and the PX version of SAEM can also be applied here.

Chapitre 6

Perspectives

Dans ce travail de thèse nous nous sommes intéressés au problème de l'estimation paramétrique dans les modèles mixtes, nous concentrant sur les modèles non–linéaires mixtes mais aussi sur les modèles linéaires généralisés mixtes (GLMM). Nous avons proposés des méthodes d'estimation pour les effets fixes et les composantes de variances dans ces modèles en nous basant sur la vraisemblance (Maximum de Vraisemblance, ML, et Maximum de Vraisemblance Restreinte, REML).

A la différence des nombreuses techniques d'estimation usuelles dans ce type de modèles, nous proposons dans ce travail des méthodes exactes en utilisant pour cela une version stochastique de l'algorithme EM, l'algorithme SAEM. Les outils standards utilisés pour l'estimation des paramètres fixes et des composants de la variance pour les modèles mixtes non–linéaires se basent sur des techniques d'approximations et de linéarisation comme par exemple le développement de Taylor de premier et second ordre. Dans le cas des GLMM, plusieurs techniques d'approximation de la fonction de vraisemblance ou du maximum de vraisemblance existent dans la littérature.
Nous avons proposé en plus de l'estimation classique ML, une nouvelle procédure d'estimation REML pour les paramètres de variance qui permet de réduire les biais associés aux estimations ML dans certains cas. Cette méthode se base sur l'interprétation bayésienne de REML qui revient à intégrer les effets fixes. L'implémentation de cette technique REML se fait avec SAEM et est assez simple à mettre en pratique aussi bien dans les modèles mixtes non–linéaires que dans les GLMM.

Toutefois, l'algorithme SAEM peut parfois présenter plusieurs inconvénients propres aux algorithmes de type EM, comme par exemple la lenteur à converger mais aussi des problèmes à atteindre le maximum global dans certaines situations. Nous proposons pour

cela une version PX de l'algorithme SAEM qui peut, parfois, résoudre ces deux problèmes, augmentant la vitesse de convergence et évitant les maxima locaux.

A l'issue de ce travail, plusieurs problèmes restent ouverts. Tout d'abord, dans le cadre de l'estimation REML, le calcul de la vraisemblance des données observées reste à travailler. Ce problème a une importance majeure puisque cela permettrait la création de tests sur les paramètres de variance. Le fait de considérer une distribution a priori impropre sur les effets fixes complique l'utilisation de la méthode d'intégration d'"Echantillonnage préférentiel" (Importance Sampling). Une possible solution à ce problème est d'utiliser un nouveau schéma adaptatif comme le "Population Monte Carlo" proposé récemment par Cappé et al. (2004). Il s'agit d'une classe de schémas d'échantillonnage préférentiel adaptatifs pour lesquels la loi instrumentale est modifiée de manière itérative en fonction de ses performances antérieures et qui, pour une fonctionnelle à intégrer donnée, minimisent la variance asymptotique de son estimateur d'importance.

Comme nous le discutons à la fin du Chapitre 5, plusieurs extensions de l'algorithme SAEM pour les GLMM sont encore possibles. Pour le cas des données binaires, en plus du modèle logistique que nous détaillons dans la section (5.8.1), il apparaît prometteur d'étudier les modèles Tobit (Tobin, 1958) très utilisés en économétrie. Ce sont des modèles pour lesquels le domaine de définition de la variable dépendante est contraint. C'est pour cela que ces modèles sont aussi appelés modèles de régression censurées ou modèles de régression tronquées. Plusieurs applications peuvent être considérées dans des domaines comme la génétique animale, la pharmacocinétique ou l'économétrie.
On peut aussi penser que l'étude des modèles qui considèrent à la fois des variables réponses continues et binaires corrélées -voir section (5.8.2)- peut encore être étendue en considérant un modèle mixte non-linéaire pour expliquer la variable continue. Cette situation pourrait parfaitement être étudiée dans le contexte de la pharmacocinétique.

Finalement, l'étude des modèles mixtes à variances hétérogènes est aussi une perspective d'étude très intéressante où l'estimation des paramètres de variance a un rôle considérable. Cette hétérogénéité peut se situer au niveau de la distribution des résidus mais aussi au niveau de la distribution des effets aléatoires. Notre procédure REML utilisant l'algorithme SAEM pourrait donc se révéler importante aussi bien dans le contexte des modèles mixtes non–linéaires que dans celui des GLMM.

Bibliographie

Agresti, A. (1999). Modeling ordered categorical data: recent advances and future challenges. *Statistics in Medicine*, 18:2191–2207.

Agresti, A. and Natarajan, R. (2001). Modeling clustered ordered categorical data: survey. *International Statistical Review*, 69:345–371.

Ashford, J. and Sowden, R. (1970). Multi-variate probit analysis. *Biometrics*, 26:535–546.

Atchley, W. and Zhu, J. (1997). Developmental quantitative genetics, conditional epigenetic variability and growth in mice. *Genetics*, 147:765–776.

Bertalanffy, L. V. (1957). Quantitative laws in metabolism and growth. *Q. Rev. Biol.*, 32:217–231.

Blasco, A., Piles, M., and Varona, L. (2003). A Bayesian analysis of the effect of selection for growth rate on growth curves in rabbits. *Genet. Sel. Evol.*, 35:21–41.

Booth, J. and Hobert, J. (1999). Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *J. Roy. Statist. Soc. B.*, 61:265–285.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *J. Amer. Statist. Assoc.*, 88:9–25.

Breslow, N. E. and Lin, X. (1995). Bias correction in generalised linear mixed models with a single component of dispersion. *Biometrika*, 82:81–91.

Brody, S. (1945). In *Bioenergetics and Growth with Special Reference to the Energetic Efficiency Complex in Domestic Animals.* Reinhold Publ., New York.

Cappé, O., Guillin, A., Marin, J.-M., and Robert, C. (2004). Population monte carlo. *J. Comput. Graph. Statist.*, 13:907–929.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm : a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational. Statistics Quaterly*, 2:73–82.

Concordet, D. and Nunez, O. G. (2002). A simulated pseudo–maximum likelihood estimator for nonlinear mixed models. *Comput. Statist. Data Anal.*, 39:187–201.

Davidian, M. and Giltinan, D. M. (2003). Nonlinear models for repeated measurements: An overview and update. *J. Agric. Biol. Env. Statist.*, 8:387–419.

Delyon, B., Lavielle, M., and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *Ann. Statist.*, 27:94–128.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39:1–38. With discussion.

Diggle, P. J., Liang, K. Y., and Zeger, S. L. (1994). *Analysis of longitudinal data.* Oxford University Press, Oxford.

Falconer, D. (1989). *An introduction to quantitative genetcis.* 3rd edition, Longman Scientific and Technical, Harlow.

Fesslern, J. A. and Hero, A. O. (1994). Space-alternating generalized expectation-maximization algorithm. *IEE Trans. Sig. Proc.*, 42:2664–2677.

Finney, D. J. (1971). *Probit Analysis.* Cambridge University Press, Cambridge University.

Fitzhugh, H. (1976). Analysis of growth curves and strategies for altering their shape. *J. Anim. Sci.*, 42:1036–1051.

Foulley, J., Gianola, D., and Thompson, R. (1983). Prediction of genetic merit from data on categorical and quantitative variates with an application to calving difficulty, birth weight and pelvic opening. *Genet. Sel. Evol.*, 15:401–424.

Foulley, J. and Manfredi, E. (1991). Approche statistique de l'évaluation de génétique des reproducteurs pour des caractères binaires à seuils. *Genet. Sel. Evol.*, 23:309–338.

Foulley, J. and Quaas, R. (1995). Heterogeneous variances in gaussian linear mixed models. *Genet. Sel. Evol.*, 27:211–228.

Foulley, J. and van Dyk, D. (2000). The PX–EM algorithm for fast stable fitting of henderson's mixed model. *Genetics Selection Evolution*, 32:143–163.

Geyer, C. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *J. R. Statist. Soc. B.*, 56:261–274.

Gianola, D. and Sorensen, D. (2006). Inferring fixed effects in a mixed linear model from an integrated likelihood. Biometrika, Submitted.

Gianola, F. (1982). Theory and analysis of threshold characters. *J. Anim. Sci.*, 54:1079–1096.

Gibbons, R. D. and Hedecker, D. (1994). Applcation of random–effects probit regression models. *Journal of Computing and Clinical Psychology*, 62:285–296.

Gilmour, A., Thompson, R., Cullis, B., and Welham, S. (2004). ASREML Manual. New South Wales Department of Agriculture, Orange, Australie.

Gueorguieva, R. V. and Agresti, A. (2001). A correlated probit model for joint modeling of clustered binary and continous responses. *J. Amer. Statist. Assoc.*, 96:1102–1112.

Harville, D. (1974). Bayesian inference for variance components using only error contrats. *Biometrika*, 61:383–385.

Hausman, J. and Wise, D. (1978). A conditional probit model for qualitative choice: Discrete decisions recognizing interdependence and heterogenous preferences. *Econometrica*, 46:403–426.

Hedeker, D. (1999). Mixno: a computer program for mixed–effects nominal logistic regression. *J. Stat. Software*, 4 (5):1–92.

Hedeker, D. and Gibbons, R. D. (1996). Mixor: a computer program for mixed-effects ordinal regression analysis. *Computer Methods and Programs in Biomedicine*, 49):157–176.

Huisman, A., Veerkamp, R., and van Arendonk, J. (2002). Genetic parameters for various random regression models to describe weight data of pigs. *J. Anim. Sci.*, 80:575–582.

Jaffrezic, F., Thompson, R., and Hill, W. (2003). Structured antedependence models for genetic analysis of multivariate repeated measures in quantitative traits. *Genet. Res.*, 82:55–65.

Jaffrezic, F., Venot, E., Lalöe, D., Vinet, A., and Renand, G. (2004). Use of structured antedependence models for the genetic analysis of growth curves. *J. Anim. Sci.*, 82:3465–3473.

Jank, W. S. (2004). Quasi–Monte Carlo sampling to improve the efficiency of Monte Carlo EM. *CSDA*, 48:685–701.

Kuhn, E. and Lavielle, M. (2004). Coupling a stochastic approximation version of EM with a MCMC procedure. *ESAIM P&S*.

Kuhn, E. and Lavielle, M. (2005). Maximum likelihood estimation in nonlinear mixed effects models. *CSDA*, 49:1020–1038.

Kung, F. (1992). Fitting logistic growth curve with predetermined carrying capacity. *Biometrics*, 48:1–17.

Laird, A. (1966). Postnatal growth of birds and mammals. *Growth*, 30:349–363.

Laird, N. M. and Ware, J. H. (1982). Random–effects models for longitudinal data. *Biometrics*, 38:963–974.

Lavielle, M. (2005). Monolix user guide manual. http://www.math.u-psud.fr/~lavielle/monolix/logiciels.

Lavielle, M. and Meza, C. (2006). A parameter expansion version of the SAEM algorithm. Statistics and Computing, Submitted.

Levine, R. A. and Casella, R. (2001). Implementations of the monte carlo EM algorithm. *J. Comp. Graph. Statist.*, 10:422–439.

Liang, K. Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, 73:13–22.

Liao, J. G. and Lipsitz, S. R. (2002). A type of restricted maximum likelihood estimator of variance components in generalised linear mixed models. *Biometrika*, 89:401–409.

Lin, X. and Breslow, N. E. (1996). Bias correction in GLMM with multiple components of dispersion. *J. Amer. Statist. Assoc.*, 91:1007–1016.

Lindstrom, M. J. and Bates, D. M. (1990). Nonlinear mixed–effects models for repeated measures data. *Biometrics*, 46:673–787.

Littell, R., Milliken, G., Stroup, W., and Wolfinger, R. (1996). *SAS System for mixed models.* SAS Institute Inc., Cary, NC, USA.

Little, R. J. A. and Rubin, D. B. (1987). *Statistical analysis with missing data.* Wiley, New York.

Liu, C., Rubin, D., and Wu, Y. (1998). Parameter expansion to accelerate EM: The PX–EM algorithm. *Biometrika*, 85:755–770.

Liu, C. and Rubin, D. B. (1994). The ECME algorithm: a simple extension of em and ecm with faster monotone convergence. *Biometrika*, 81:633–648.

Louis, T. (1982). Finding the observed information matrix when using the em algorithm. *J. Roy. Statist. Soc. B.*, 44:226–233.

Ma, C.-X., Casella, G., and Wu, R. (2002). Functional mapping of quantitative trait loci underlying the character process: a theoretical framework. *Genetics*, 161:1751–1762.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models.* Chapman and Hall, New York.

McCulloch, C. E. (1997). Maximum likelihood algorithms for generalized linear mixed models. *J. Amer. Statist. Assoc.*, 92:162–170.

McFadden, D. (1989). A method of simulated moments for estimation of discrete repsonse models without numerical integration. *Econometrica*, 57:995–1026.

McLachlan, G. J. and Krishnan, T. (1997). *The EM algorithm and extensions.* J. Wiley and Sons, New York.

Meng, X. and van Dyk, D. (1998). Fast em implementations for mixed-effects models. *J. Roy. Statist. Soc. B*, 60:559–578.

Meng, X.-L. and Rubin, D. B. (1993a). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika*, 80(2):267–278.

Meng, X.-L. and Rubin, D. B. (1993b). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika*, 80 (2):267–278.

Meza, C., Jaffrézic, F., and Foulley, J. L. (2006). REML estimation of variance parameters in nonlinear mixed effects models using the SAEM algorithm. Biometrical Journal, Submitted,.

Mialon, M. M., Renand, G., Krauss, D., and Ménissier, F. (2001). Variability of the postpartum recovery of sexual activity of Charolais cows. *Livest. Prod. Sci.*, 69:217–226.

Mignon-Grasteau, S., Piles, M., Varona, L., Poivey, J. P., et al. (2000). Genetic analysis of growth curve parameters for male and female chickens resulting from selection on shape of growth curve. *J. Anim. Sci.*, 78:2532–2531.

Molenberghs, G. and Verbeke, G. (2005). *Models for Discrete Longitudinal Data.* Springer, New York.

Nelder, J. A. and Lee, Y. (1992). Likelihood, quasi-likelihood and pseudolikelihood: Some comparisons. *J. Roy. Statist. Soc. B*, 54 (1):273–284.

Neuhaus, J. and Segal, M. (1997). An assessment of approximate maximum likelihood estimators in generalized linear models. In Gregoire, T. G., editor, *Modelling longitudinal and spatially correlated data: Methods, Applications and future directions. Lecture notes in Statistics, v. 122.* Springer, New York.

Nunez-Anton, V. and Zimmerman, D. (2000). Modeling non-stationary longitudinal data. *Biometrics*, 56:699–705.

Patterson, H. and Thompson, R. (1971). Recovery of interblock information when block sizes are unequal. *Biometrika*, 58:545–554.

Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-effects models in S and S-Plus.* Statistics and Computing, Springer, New York.

Pletcher, S. D. and Jaffrézic, F. (2002). Generalized character process models: estimating the genetic basis of traits that cannot be observed and that change with age or environmental conditions. *Biometrics*, 58:157–162.

Raudenbush, S. W., Yang, M.-L., and Yosef, M. (2000). Maximum likelihood for generalized linear models with nested random effects via high–order, multivariate laplace approximation. *J. Comp. Graph. Stat.*, 9:141–157.

Robert, C. P. (1995). Simulation of truncated normal variables. *Statistics and Computing*, 5:121–125.

Roberts, G. O., Gelman, A., and Gilks, W. (1997). Weak convergence and optimal scaling of random walk metropolis algorithm. *Ann. Applied Prob.*, 7:110–120.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling of various metropolis–hastings algorithms. *Statistical Science*, 16:351–367.

Rodríguez, G. and Goldman, N. (1995). An assessment of estimation procedures for multilevel models whitn binary responses. *J. Roy. Statist. Soc. A*, 158:73–89.

Schall, R. (1991). Estimation in generalized linear models with random effects. *Biometrika*, 78:719–727.

Spiegelhalter, D., Thomas, A., and Best, N. (2004). Winbugs 1.4 user manual. Cambridge: Medical Research Council Biostatistics Unit. http://www.mrc-bsu.cam.ac.uk/bugs.

Thall, P. and Vail, S. (1990). Some covariance models for longitudinal count data with ovserdispersion. *Biometrics*, 46):657–671.

Tobin, J. (1958). Estimation of relationships for limited dependent variables. *Econometrica*, 26:24–36.

Verbeke, G. and Molenberghs (2000). *Linear mixed models for longitudinal data.* Springer–Verlag, New York.

Vonesh, E. and Carter, R. (1992). Mixed–effects non linear regression for unbalanced repeated measures. *Biometrics*, 48:1–17.

Wei, G. and Tanner, M. (1990a). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J. Amer. Statist. Assoc.*, 85:699–704.

Wei, G. and Tanner, M. (1990b). A monte carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J. Amer. Statist. Assoc.*, 85:699–704.

Wolfinger, R. (1992). Laplace's approximation for nonlinear mixed models. *Biometrika*, 80:791–795.

Wu, C.-F. J. (1983). On the convergence properties of the EM algorithm. *Ann. Statist.*, 11:95–103.

Wu, L. (2004). Exact and approximate inferences for nonlinear mixed-effects models with missing covariates. *J. Amer. Statist. Assoc.*, 99:700–709.