J. R. Statist. Soc. B (1998)
60, *Part* 3, *pp.* 559–578

Fast EM-type implementations for mixed effects models

Xiao-Li Meng†
University of Chicago, USA

and David van Dyk
Harvard University, Cambridge, USA

[Received February 1996. Final revision September 1997]

Summary. The mixed effects model, in its various forms, is a common model in applied statistics. A useful strategy for fitting this model implements EM-type algorithms by treating the random effects as missing data. Such implementations, however, can be painfully slow when the variances of the random effects are small relative to the residual variance. In this paper, we apply the 'working parameter' approach to derive alternative EM-type implementations for fitting mixed effects models, which we show empirically can be hundreds of times faster than the common EM-type implementations. In our limited simulations, they also compare well with the routines in S-PLUS® and Stata® in terms of both speed and reliability. The central idea of the working parameter approach is to search for efficient data augmentation schemes for implementing the EM algorithm by minimizing the augmented information over the working parameter, and in the mixed effects setting this leads to a transfer of the mixed effects variances into the regression slope parameters. We also describe a variation for computing the restricted maximum likelihood estimate and an adaptive algorithm that takes advantage of both the standard and the alternative EM-type implementations.

Keywords: Data augmentation; Incomplete data; Missing data; Random effects models; Rate of convergence; Restricted maximum likelihood; Variance components models

1. Introduction

Since Dempster *et al.* (1977) showed its great potential for finding maximum likelihood estimates or more generally posterior modes, in problems that involve missing data or can be formulated as such, the EM algorithm has become one of the most well-known and used techniques in applied statistics. Fitting mixed effects models with the EM algorithm was one of the more novel applications presented in Dempster *et al.* (1977) because this particular application makes it clear that we can apply the algorithm even when there are no missing data in the usual sense, in that we can treat latent variables as missing values. Since Dempster *et al.* (1977) this topic has been well developed in the literature (e.g. Laird (1982), Laird and Ware (1982), Dempster *et al.* (1984), Laird *et al.* (1987) and Liu and Rubin (1994)). In particular a series of articles on animal breeding studies which use variance component models fitted via the EM algorithm has made the *Journal of Dairy Science* fourth in the list of journals that have published the most EM-related articles since Dempster *et al.* (1977), according to Meng and Pedlow (1992) (also see Meng and van Dyk (1997)).

†*Address for correspondence*: Department of Statistics, University of Chicago, 5734 University Avenue, Chicago, IL 60637, USA.
E-mail: meng@galton.uchicago.edu

1369–7412/98/60559

Although the simplicity and stability (e.g. monotone convergence in likelihood values) of the EM algorithm have made it a popular method for fitting mixed effects models, the algorithm's slow convergence, especially when the variances of the random effects are relatively small, has led various researchers to consider alternatives (e.g. Thompson and Meyer (1986), Lindstrom and Bates (1988) and Callanan and Harville (1991); also see Harville (1977)). Although these alternatives (e.g. Newton–Raphson iteration) do provide fast convergence with careful implementation and monitoring, they have not been as popular as the EM algorithm in practice mainly because they require a larger human effort. For example, without extra computational effort, these algorithms can produce negative variance estimates (e.g. Thompson and Meyer (1986) and Callanan and Harville (1991)). Even with careful monitoring, as implemented by some commercial software, such algorithms can converge to a wrong point *within* the parameter space. For example, we have encountered such cases in our use of S-PLUS, as reported in Section 3.2. It is therefore of practical interest to speed up EM-type algorithms within the EM framework, and the 'working parameter' method of Meng and van Dyk (1997) provides one way of searching for fast EM-type implementations.

Briefly, the EM algorithm and its various extensions start by defining an augmentation Y_{aug} such that the observed data $Y_{\text{obs}} = \mathcal{M}(Y_{\text{aug}})$ for some many-to-one mapping $\mathcal{M}$ (more precisely, Y_{aug} is shorthand for the underlying augmented data model). The theoretical speed of convergence of the algorithm is then determined by the smallest eigenvalue of the matrix 'fraction of observed information' (Dempster *et al.*, 1977) $I_{\text{obs}} I_{\text{aug}}^{-1}$, where I_{aug} is the expected augmented Fisher information matrix and I_{obs} is the observed Fisher information matrix (see Meng and van Dyk (1997) for details). The key idea of the approach adopted in Meng and van Dyk (1997) is to construct a class of augmentations, $Y_{\text{aug}}(a)$, indexed by a working parameter a such that $Y_{\text{obs}} = \mathcal{M}_a\{Y_{\text{aug}}(a)\}$; here the mapping $\mathcal{M}_a$ can depend on a. Once a class of possible data augmentation schemes has been constructed, we can search for optimal or nearly optimal values of a in terms of minimizing the expected augmented Fisher information I_{aug}, and thus maximizing the theoretical speed of convergence; note that I_{obs} does not depend on a. Constructing $Y_{\text{aug}}(a)$ such that the resulting algorithm is not only fast but also easy to implement is a matter of art in that, just as in the actual EM implementation, it needs to be worked out case by case. Meng and van Dyk (1997) illustrated how this approach can be used to construct efficient EM implementations for fitting multivariate t-models and for image reconstruction under a Poisson model. Here we present another application: the mixed effects model.

Our presentation is organized as follows. After a brief review of the standard EM and EM-type implementations, Section 2 presents EM-type implementations based on our new data augmentation scheme; we also discuss the computation for restricted maximum likelihood (REML). Section 3 presents empirical evidence that a sensible selection of the data augmentation scheme can lead to EM-type implementations that are dramatically faster than the conventional implementations (e.g. more than 100 times faster) and comparable with some commercially available routines but with preferable convergence properties. Section 4 provides theoretical insights using a simple mixed effects model. Section 5 concludes with a brief remark on the potential generalization to the Gibbs sampler.

2. Standard and alternative implementations

2.1. The standard implementation

We consider the mixed effects model of the form

$$y_i = X_i^{\mathrm{T}}\beta + Z_i^{\mathrm{T}} b_i + e_i, \qquad b_i \sim N_q(0, T), \quad e_i \sim N(0, \sigma^2 R_i), \quad b_i \perp e_i, \tag{2.1}$$

for $i = 1, \ldots, m$, where the (observed) response y_i is $n_i \times 1$, X_i ($p \times n_i$) and Z_i ($q \times n_i$) are known covariates (throughout we assume that the Z_i are such that T is identifiable), β are the $p \times 1$ fixed effects, $b_i = (b_{i1}, \ldots, b_{iq})^{\mathrm{T}}$ are the $q \times 1$ random effects and $R_i > 0$ are known $n_i \times n_i$ matrices. (We follow the literature to use R_i explicitly, though mathematically it is not more general than $R_i = I$ since it is assumed to be known.) Although there is no general closed form solution for the maximum likelihood estimate $\theta^* \equiv (\beta^*, \sigma^{2*}, T^*)$ of $\theta \equiv (\beta, \sigma^2, T)$ given $Y_{\mathrm{obs}} = \{(y_i, X_i, Z_i, R_i), i = 1, \ldots, m\}$, the EM algorithm provides a simple and stable fitting algorithm. The standard data augmentation (Dempster *et al.*, 1977; Laird and Ware, 1982; Laird *et al.*, 1987) which treats the b_i as missing data (i.e. $Y_{\mathrm{aug}} = \{(y_i, X_i, Z_i, R_i, b_i), i = 1, \ldots, m\}$) leads naturally to the following algorithm. The E-step finds the conditional expectation of the log-likelihood function of θ based on the augmented data Y_{aug}, conditionally on Y_{obs} and $\theta^{(t)}$ from the previous iteration of the algorithm, $Q(\theta|\theta^{(t)}) = E\{l(\theta|Y_{\mathrm{aug}})|Y_{\mathrm{obs}}, \theta^{(t)}\}$. This amounts to calculating $E(b_i|Y_{\mathrm{obs}}, \theta^{(t)})$ and $E(b_i b_i^{\mathrm{T}}|Y_{\mathrm{obs}}, \theta^{(t)})$, $i = 1, \ldots, m$, which can be obtained easily because

$$b_i|Y_{\mathrm{obs}}, \theta \sim N_q\{\hat{b}_i(\theta), T - TZ_i W_i(\zeta) Z_i^{\mathrm{T}} T\}, \qquad i = 1, \ldots, m, \tag{2.2}$$

where

$$\hat{b}_i(\theta) = TZ_i W_i(\zeta)(y_i - X_i^{\mathrm{T}}\beta) \qquad \text{with } W_i(\zeta) = (\sigma^2 R_i + Z_i^{\mathrm{T}} T Z_i)^{-1}, \quad \zeta = (\sigma^2, T). \tag{2.3}$$

The computation of $W_i(\zeta)$ can be facilitated by reducing the dimension of the matrix inversion when $q < n_i$ by using the matrix identity (R_i^{-1} is only computed once since it does not change with iteration):

$$W_i(\zeta) = \{R_i^{-1} - R_i^{-1} Z_i^{\mathrm{T}} (\sigma^2 T^{-1} + Z_i R_i^{-1} Z_i^{\mathrm{T}})^{-1} Z_i R_i^{-1}\}/\sigma^2.$$

The M-step of the algorithm then updates θ with the value $\theta^{(t+1)}$ which maximizes the expected augmented data log-likelihood $Q(\theta|\theta^{(t)})$, which in this case factors into two terms, one involving β and σ^2 and the other involving T, and thus the M-step has a particularly simple form. First we update (β, σ^2) via the linear regression implied by model (2.1),

$$\beta^{(t+1)} = \left(\sum_{i=1}^{m} X_i R_i^{-1} X_i^{\mathrm{T}}\right)^{-1} \sum_{i=1}^{m} X_i R_i^{-1}\{y_i - Z_i^{\mathrm{T}}\, \hat{b}_i(\theta^{(t)})\} \tag{2.4}$$

and, letting $r_i^{(t+1)} = y_i - X_i^{\mathrm{T}} \beta^{(t+1)} - Z_i^{\mathrm{T}} \hat{b}_i(\theta^{(t)})$,

$$\sigma^{2(t+1)} = \frac{1}{n} \sum_{i=1}^{m} [r_i^{(t+1)\mathrm{T}} R_i^{-1} r_i^{(t+1)} + \mathrm{tr}\{R_i^{-1} \operatorname{var}(Z_i^{\mathrm{T}} b_i|Y_{\mathrm{obs}}, \theta^{(t)})\}] \tag{2.5a}$$

$$= \frac{1}{n} \sum_{i=1}^{m} [r_i^{(t+1)\mathrm{T}} R_i^{-1} r_i^{(t+1)} + \sigma^{2(t)}\, \mathrm{tr}\{I - \sigma^{2(t)}\, W_i(\zeta^{(t)}) R_i\}], \tag{2.5b}$$

where $n = \Sigma_{i=1}^{m} n_i$; we express $\sigma^{2(t+1)}$ in two ways here to facilitate the discussion of REML computations in Section 2.3. We then update T with

$$T^{(t+1)} = \frac{1}{m} \sum_{i=1}^{m} E(b_i b_i^{\mathrm{T}}|Y_{\mathrm{obs}}, \theta^{(t)}) = \frac{1}{m} \sum_{i=1}^{m} \hat{b}_i(\theta^{(t)})\, \hat{b}_i(\theta^{(t)})^{\mathrm{T}} + T^{(t)} - T^{(t)} \left\{\frac{1}{m} \sum_{i=1}^{m} Z_i W_i(\zeta^{(t)}) Z_i^{\mathrm{T}}\right\} T^{(t)}, \tag{2.6}$$

thus completing a single iteration of the standard EM implementation.

It is generally advisable (e.g. Laird and Ware (1982) and Laird *et al.* (1987)) to replace equation (2.4) with

$$\beta^{(t+1)} = \left\{ \sum_{i=1}^{m} X_i W_i(\zeta^{(t+1)}) X_i^{\mathrm{T}} \right\}^{-1} \sum_{i=1}^{m} X_i W_i(\zeta^{(t+1)}) y_i, \tag{2.7}$$

the conditional maximizer of $l(\theta|Y_{\mathrm{obs}})$ with $\zeta = (T, \sigma^2)$ fixed at its most recent update $\zeta^{(t+1)}$. Accordingly, the $\beta^{(t+1)}$ in $r_i^{(t+1)}$ needs to be replaced by $\beta^{(t)}$ when computing equation (2.5a) or (2.5b). This replacement can be justified by the theory underlying the 'expectation–conditional maximization either' (ECME) algorithm (Liu and Rubin, 1994), which allows maximizing *either* the expected log-likelihood $Q(\theta|\theta^{(t)})$ or the actual log-likelihood $l(\theta|Y_{\mathrm{obs}})$. More generally, it is an example of the 'alternating expectation–conditional maximization' algorithm AECM that we discussed in Meng and van Dyk (1997). We use $\zeta^{(t+1)}$ in equation (2.7) instead of $\zeta^{(t)}$ because of an order restriction on implementing the ECME algorithm, as discussed in Meng and van Dyk (1997). More specifically, with the ECME algorithm, equations (2.5a)–(2.6) form the first CM-step, which maximizes $Q(\theta|\theta^{(t)})$ conditionally on $\beta = \beta^{(t)}$, and equation (2.7) gives the second CM-step, which maximizes $l(\theta|Y_{\mathrm{obs}})$ conditionally on $\zeta = \zeta^{(t+1)}$, the output of the first CM-step. An interesting by-product of this ECME implementation is that it unifies maximum likelihood and REML computations, as we shall discuss in Section 2.3. For clarity, we shall distinguish between ECME and EM, whenever appropriate.

2.2. Alternative implementations

To search for an efficient EM algorithm for fitting t-models, Meng and van Dyk (1997) introduced a working parameter into the data augmentation scheme by rescaling the missing variable which resulted in a remarkably fast EM implementation for the t-model. Inspired by this success, we tried the same idea with the mixed effects model (2.1). Because in this setting the unobserved random variable b is generally a vector, the construction of an appropriate data augmentation scheme is more complicated. In principle, we can rescale b by $T^{-a/2}$, where a is an arbitrary constant, and treat $\{b_i(a) = T^{-a/2} b_i,\ i = 1, \ldots, n\}$ as the missing data. Indeed, when b_i is univariate or when T is assumed to be proportional to a known matrix, this rescaling with $a = 1$ is a natural consequence of expressing model (2.1) with standardized random effects, as in Anderson and Aitkin (1985) with binary response and in Foulley and Quass (1995) with heterogeneous variances mixed effects models. For a general T, however, it is difficult, if not impossible, to implement the EM or ECME algorithms resulting from using $T^{-a/2}$ with an arbitrary a. This violates our requirement that the resulting algorithm not only needs to be fast but also needs to be simple and stable. For a discussion of the resulting algorithms in the special case of $a = 0$ or $a = 1$, however, see van Dyk (1995).

To circumvent this problem, we use the Choleski decomposition to diagonalize (i.e. to orthogonalize) T before we implement an EM or ECME algorithm. Specifically, we let $T = \Delta U \Delta^{\mathrm{T}}$, where Δ is a lower triangular $q \times q$ matrix with 1s on the diagonal and U is a diagonal matrix. It is well known that such a decomposition exists and is unique (e.g. Horn and Johnson (1985), p. 162). This parameterization of T has been used to help to stabilize Newton–Raphson-type algorithms (e.g. Lindstrom and Bates (1988) and Groeneveld (1994)) and for reducing computation time within each iteration of the standard EM or ECME implementation (Lindstrom and Bates, 1988). We use it to introduce a new class of data augmentation schemes by letting $c_i = \Delta^{-1} b_i$, then $c_i \sim N_q(0, U)$. Since $U \equiv \mathrm{diag}\{u_1^2, \ldots, u_q^2\}$ is diagonal, we now have the flexibility to rescale each element of $c_i = (c_{i1}, \ldots, c_{iq})^{\mathrm{T}}$ by a power of its own standard deviation. Specifically, for any vector $a = (a_1, \ldots, a_q)^{\mathrm{T}} \in \mathbf{R}^q$, we can define

$$c_i(a) = \left(\frac{c_{i1}}{u_1^{a_1}}, \frac{c_{i2}}{u_2^{a_2}}, \ldots, \frac{c_{iq}}{u_q^{a_q}} \right)^{\mathrm{T}}$$

and treat $Y_{\mathrm{aug}}(a) = \{(y_i, X_i, Z_i, R_i, c_i(a)), i = 1, \ldots, n\}$ as the augmented data. Notice that the definition of $c_i(a)$ depends on the order of the random effects and thus there are $q!$ possible data augmentation schemes of this sort. We do not consider the issue of ordering here, which may be worthy of investigation, though we doubt that it has a substantial effect in typical applications.

With this alternative augmentation, model (2.1) can be expressed as

$$y_i = X_i^{\mathrm{T}}\beta + Z_i^{\mathrm{T}}\Delta\tilde{U}(a)c_i(a) + e_i = X_i^{\mathrm{T}}\beta + \sum_{j=1}^{q}\sum_{k=j}^{q} c_{ij}(a)Z_{ik}\delta_{kj}u_j^{a_j} + e_i, \tag{2.8}$$

where $\Delta = (\delta_{kj})$, $\tilde{U}(a) \equiv \mathrm{diag}\{u_1^{a_1}, \ldots, u_q^{a_q}\}$, $c_i(a) = (c_{i1}(a), \ldots, c_{iq}(a))^{\mathrm{T}}$ and $Z_i = (Z_{i1}, \ldots, Z_{iq})^{\mathrm{T}}$ with Z_{ik} an $n_i \times 1$ vector. Although in principle we can derive the EM algorithm for any $a \in \mathbf{R}^q$, we restrict ourselves in this paper to $a \in \{0, 1\}^q$, i.e. a_i can only take values 0 or 1, to keep the resulting algorithms simple to implement (i.e. a closed form M-step), which is one of the main objectives of our search; a more general search can lead to even faster algorithms but can also increase the complexity of the algorithms. Within this class of data augmentation schemes, given $Y_{\mathrm{aug}}(a)$, model (2.8) is a linear regression with $p + q(q-1)/2 + \Sigma_j^q a_j$ regression coefficients when we view (note that $\delta_{jj} = 1$ for all j)

$$\{\delta_{kj}u_j, k \geqslant j, \text{ for } a_j = 1\} \cup \{\delta_{kj}; k > j, \text{ for } a_j = 0\}$$

as the $q(q-1)/2 + \Sigma_j^q a_j$ regression coefficients besides β. Notice that model (2.8) is a regression on β and the elements of Δ when $a = (0, \ldots, 0)^{\mathrm{T}}$; when $a = (1, \ldots, 1)^{\mathrm{T}}$, in contrast, model (2.8) is a regression on β and the elements of the lower triangular matrix $L = \Delta U^{1/2}$ (i.e. $T = LL^{\mathrm{T}}$).

As discussed at the end of Section 2.1, the ECME implementation has two CM-steps at each iteration. For our new augmentation scheme, the first CM-step updates (σ^2, Δ, $\{u_j$, for $a_j = 1\}$) via the linear regression (2.8), by treating $\{c_{ij}(a)Z_{ik}, k \geqslant j\}$ as the missing covariates and $\beta = \beta^{(t)}$. For example, for $a = (1, \ldots, 1)^{\mathrm{T}}$ (this is the case that we generally recommend, as we shall reason later), we can rewrite model (2.8) as

$$y_i - X_i^{\mathrm{T}}\beta^{(t)} = \tilde{X}_i^{\mathrm{T}}\tilde{\beta} + e_i,$$

where $\tilde{X}_i$ is a $[q(q+1)/2] \times n_i$ matrix with rows $c_{ij}(a)Z_{ik}^{\mathrm{T}}$ for $j = 1, \ldots, q$, $k = j, \ldots, q$, and $\tilde{\beta}$ is a vector with corresponding components $\delta_{kj}u_j$. Consequently, (σ^2, Δ, U) is updated by

$$\tilde{\beta}^{(t+1)} = \left\{\sum_{i=1}^{m} \tilde{B}_i(\theta^{(t)})\right\}^{-1} \sum_{i=1}^{m} \tilde{X}_i(\theta^{(t)})R_i^{-1}(y_i - X_i^{\mathrm{T}}\beta^{(t)}) \tag{2.9}$$

and, letting $\tilde{r}_i^{(t)} = y_i - X_i^{\mathrm{T}}\beta^{(t)} - Z_i^{\mathrm{T}}\Delta^{(t)}\tilde{U}^{(t)}(a)\hat{c}_i(a, \theta^{(t)})$,

$$\sigma^{2(t+1)} = \frac{1}{n}\sum_{i=1}^{m}[\tilde{r}_i^{(t)\mathrm{T}}R_i^{-1}\tilde{r}_i^{(t)} + \sigma^{2(t)}\,\mathrm{tr}\{I - \sigma^{2(t)}W_i(\zeta^{(t)})R_i\}], \tag{2.10}$$

where $\tilde{X}_i(\theta) = E(\tilde{X}_i | Y_{\mathrm{obs}}, \theta)$, $\tilde{B}_i(\theta) = E(\tilde{X}_i R_i^{-1}\tilde{X}_i^{\mathrm{T}} | Y_{\mathrm{obs}}, \theta)$ and $\hat{c}_i(a, \theta) = E\{c_i(a) | Y_{\mathrm{obs}}, \theta\}$, which are found in the E-step (see below); note here that $W_i(\zeta)$ is the same as in equation (2.3) except that T is calculated as $\Delta U \Delta^{\mathrm{T}}$. Computationally, the matrix inversion in equation (2.9) can be avoided by using the SWEEP operator (Beaton, 1964), as discussed in Little and Rubin (1987), pages 53–57; for $R_i \neq I_{n_i}$, a simple orthogonalization (i.e. $R_i^{-1/2} X_i^{\mathrm{T}}$ and $R_i^{-1/2} Z_i^{\mathrm{T}}$) may need to be performed before iterations. Finally, for general $a \in \{0, 1\}^q$, we update $\{u_j$, for $a_j = 0\}$ by using $c_{ij}(a) \sim N(0, u_j^2)$ when $a_j = 0$ and thus

$$u_j^{2(t+1)} = \frac{1}{m} \sum_{i=1}^{m} E\{c_{ij}^2(a) | Y_{\mathrm{obs}}, \theta^{(t)}\}, \qquad \text{for } j \text{ such that } a_j = 0. \tag{2.11}$$

When $a = (1, \ldots, 1)^{\mathrm{T}}$, equation (2.11) is not needed. The second CM-step is the same as equation (2.7).

To perform the E-step, first we note that

$$\hat{c}_i(a, \theta) = \tilde{U}(2 - a) \Delta^{\mathrm{T}} Z_i W_i(\zeta)(y_i - X_i^{\mathrm{T}} \beta), \tag{2.12}$$

recall that $\tilde{U}(a) = \mathrm{diag}\{u_1^{a_1}, \ldots, u_q^{a_q}\}$ and

$$\begin{aligned} \hat{B}_i(a, \theta) &= E\{c_i(a) c_i^{\mathrm{T}}(a) | Y_{\mathrm{obs}}, \theta\} \\ &= \hat{c}_i(a, \theta) \hat{c}_i^{\mathrm{T}}(a, \theta) + \tilde{U}\{2(1 - a)\} - \tilde{U}(2 - a) \Delta^{\mathrm{T}} Z_i W_i(\zeta) Z_i^{\mathrm{T}} \Delta \tilde{U}(2 - a), \end{aligned} \tag{2.13}$$

where we have used the fact that $U = \tilde{U}(2)$ and $2 - a$ means $(2 - a_1, \ldots, 2 - a_q)^{\mathrm{T}}$, etc. We then use the components of $\hat{c}_i(a, \theta^{(t)})$ and the elements of $\hat{B}_i(a, \theta^{(t)})$, $i = 1, \ldots, n$, to calculate the conditional expectations of the required augmented data sufficient statistics. In particular, $E\{c_{ij}^2(a) | Y_{\mathrm{obs}}, \theta^{(t)}\}$ needed for equation (2.11) is simply the (j, j)th (diagonal) element of $\hat{B}_i(a, \theta^{(t)})$. The rows of $\tilde{X}_i(\theta^{(t)})$ are calculated with

$$E\{c_{ij}(a) Z_{ik}^{\mathrm{T}} | Y_{\mathrm{obs}}, \theta^{(t)}\} = \hat{c}_{ij}(a, \theta^{(t)}) Z_{ik}^{\mathrm{T}}, \tag{2.14}$$

for $j = 1, \ldots, q$ and $k \geqslant j$, where $\hat{c}_{ij}(a, \theta^{(t)})$ is the jth component of the vector $\hat{c}_i(a, \theta^{(t)})$. The elements of $\tilde{B}_i(\theta^{(t)})$ are calculated using

$$E\{c_{ij}(a) Z_{ik}^{\mathrm{T}} R_i^{-1} Z_{im} c_{il}(a) | Y_{\mathrm{obs}}, \theta^{(t)}\} = [\hat{B}_i(a, \theta^{(t)})]_{jl} Z_{ik}^{\mathrm{T}} R_i^{-1} Z_{im}, \tag{2.15}$$

for $j = 1, \ldots, q$, $k \geqslant j$, $l = 1, \ldots, q$ and $m \geqslant l$, where $[\hat{B}_i(a, \theta^{(t)})]_{jl}$ is the (j, l)th element of $\hat{B}_i(a, \theta^{(t)})$.

Once the algorithm has converged, it is easy to compute the original parameter via $T^* = \Delta^* U^* \Delta^{*\mathrm{T}}$. Fitting the regression model (2.8) can result in negative values for the $\{u_j^*, j = 1, \ldots, q\}$ (in certain special cases, Foulley and Quass (1995) have shown that the negative values cannot occur as long as the initial values are positive). Concerns for this possibility (e.g. Thompson (1995)) should be distinguished from the concerns of negative variance estimates from using algorithms like the Newton–Raphson algorithm, for which a negative estimate indicates a numerical error and one generally cannot recover the correct estimate from it. In our case, the recovery is trivial because $\Delta^* U^* \Delta^{*\mathrm{T}}$ will remain positive semidefinite regardless of the sign of the componens of $(u_1, \ldots, u_q)$. In fact, since Δ and U are unique for each T, there are exactly $2^{\Sigma_j a_j}$ modes of $l(\beta, \sigma^2, \tilde{U}(1), \Delta | Y_{\mathrm{obs}})$ (corresponding to the diagonal roots of U) for every mode of $l(\beta, \sigma^2, T | Y_{\mathrm{obs}})$. In other words, when we rewrite model (2.1) as model (2.8), it is understood that the support of each u_j has been extended to the whole real line when $a_j = 1$. The extension of the support of $(u_1, \ldots, u_q)$ seems to be one reason why the alternative algorithms can be much faster; see Section 3.4 for more discussion.

We now have $2^q + 1$ algorithms when $q > 1$, i.e. the 2^q algorithms corresponding to $a \in \{0, 1\}^q$ and the standard ECME implementation, which corresponds to $a = (0, \ldots, 0)^{\mathrm{T}}$ if Δ is constrained to the identity matrix in the fitted model; for $q = 1$, the standard algorithm is the same as the new algorithm with $a = 0$. Each of these algorithms is straightforward to implement but they will generally converge at different speeds. To evaluate the relative computational merit of the algorithms we shall present simulation studies in Section 3 and outline theoretical comparisons in Section 4, after we discuss some useful variations of the algorithms in Section 2.3.

2.3. Variations

Variations of the algorithms can be derived to accommodate structure in T. For example, suppose that $T = \tau^2 B$, where $B = LL^{\mathrm{T}}$ is known with L lower triangular. In this case, we can write the model as

$$y_i = X_i^{\mathrm{T}} \beta + \tau^a Z_i^{\mathrm{T}} L c_i(a) + e_i,$$

where $c_i(a) = L^{-1} b_i / \tau^a$ for $a \in \{0, 1\}$. When $a = 1$, τ can be updated as a regression parameter with missing covariate $Z_i^{\mathrm{T}} L c_i(a)$. Similar calculations lead to algorithms for T block diagonal where each block can be completely unknown, completely known or known up to a scale factor. An algorithm similar to ours with $a = (1, \ldots, 1)^{\mathrm{T}}$ (but without using the Choleski decomposition) for the special case of T block diagonal with each block known up to a scale factor was presented by Foulley and Quass (1995). They, however, suggested the use of Newton–Raphson iteration when T is not block diagonal or when some blocks are completely unknown.

The algorithm described in Section 2.2 can also be modified for REML computations, which corresponds to an empirical Bayesian analysis with the constant (improper) prior on β, as detailed in Laird and Ware (1982). The variation is obtained by replacing $\theta = (\beta, \sigma^2, T)$ with $\zeta = (\sigma^2, T) = (\sigma^2, \Delta, U)$ as the parameter of interest and including β in the data augmentation, i.e. by treating β as missing data, with an improper uniform distribution on $\mathbf{R}^p$. Note that the EM theory (e.g. Dempster *et al.* (1977) and Wu (1983)) does not require that $f(Y_{\mathrm{aug}} | \theta)$ be a proper density, but only that $f(Y_{\mathrm{mis}} | Y_{\mathrm{obs}}, \theta)$ be proper so that the E-step is well defined. This point, which seems not to be generally well recognized, is important because improper distributions are frequently encountered in Bayesian computations; see the rejoinder to Meng and van Dyk (1997) for additional discussion.

Once we realize that we can perform REML calculations by using the EM algorithm and treating β as additional missing data, we can easily modify the algorithms for maximum likelihood to accomplish our computational goal. For REML, we only need to replace $W_i(\zeta)$ in equation (2.5b) and in equation (2.6) by (see Laird *et al.* (1987))

$$P_i(\zeta) = W_i(\zeta) - W_i(\zeta) X_i^{\mathrm{T}} \left\{ \sum_{i=1}^{m} X_i W_i(\zeta) X_i^{\mathrm{T}} \right\}^{-1} X_i W_i(\zeta).$$

This replacement is the consequence of the additional E-step calculations for finding the conditional expectations of the required missing augmented data sufficient statistics, which now includes β, $\beta\beta^{\mathrm{T}}$ and $\{\beta b_i^{\mathrm{T}}, i = 1, \ldots, m\}$ in addition to $\{b_i, b_i b_i^{\mathrm{T}}, i = 1, \ldots, m\}$. In particular, the term, $\mathrm{var}(Z_i^{\mathrm{T}} b_i | Y_{\mathrm{obs}}, \theta^{(t)})$ in equation (2.5a) needs to be replaced by $\mathrm{var}(X_i^{\mathrm{T}} \beta + Z_i^{\mathrm{T}} b_i | Y_{\mathrm{obs}}, \zeta^{(t)})$ because β is now part of the augmented data. The required additional derivation is straightforward by combining expression (2.2) with the fact that under a constant prior on β

$$\beta | Y_{\mathrm{obs}}, \zeta \sim N\left[\hat{\beta}(\zeta), \left\{\sum_{i=1}^{m} X_i W_i(\zeta) X_i^{\mathrm{T}}\right\}^{-1}\right],$$

where $\hat{\beta}(\zeta)$ has the same expression as equation (2.7) with $\zeta^{(t)}$ replaced by ζ.

It is worthwhile to point out that, although expression (2.7) stays the same for both maximum likelihood and REML calculations, it has different interpretations in the two calculations. For the maximum likelihood calculation, equation (2.7) comes from an (conditional) M-step, since β is a parameter in the sampling model. For the REML calculation, however, equation (2.7) comes from the E-step since β is a part of the augmented data, just like the random effects, $\{b_1, \ldots, b_m\}$. That these two steps produce the same expression is due to the fact that for a normal distribution the mean (i.e. the E-step) is the same as the mode (i.e. the M-step).

For the alternative implementation described in Section 2.2, we can carry out a similar, albeit less straightforward, modification for the REML computations. In addition to replacing $W_i(\zeta)$ with $P_i(\zeta)$ in equations (2.10) and (2.13) we need to replace equation (2.9) with

$$\tilde{\beta}^{(t+1)} = \left\{\sum_{i=1}^{m} \tilde{B}_i(\theta^{(t)})\right\}^{-1} E\left\{\sum_{i=1}^{m} \tilde{X}_i R_i^{-1}(y_i - X_i^{\mathrm{T}}\beta) \middle| Y_{\mathrm{obs}}, \zeta^{(t)}\right\}, \qquad (2.16)$$

where $\tilde{B}_i(\theta^{(t)})$ is *calculated* in the same way as before, using equation (2.15) with $\theta^{(t)} = (\hat{\beta}(\zeta^{(t)}), \zeta^{(t)})$ and noting the replacement in equation (2.13). The second term on the right-hand side of equation (2.16) can be computed via

$$E\{c_{ij}(a) Z_{ik}^{\mathrm{T}} R_i^{-1}(y_i - X_i^{\mathrm{T}}\beta) | Y_{\mathrm{obs}}, \zeta^{(t)}\} = \hat{c}_{ij}(a, \theta^{(t)}) Z_{ik}^{\mathrm{T}} R_i^{-1} y_i - Z_{ik}^{\mathrm{T}} R_i^{-1} X_i^{\mathrm{T}} [\hat{D}_i(\zeta^{(t)})]_j^{\mathrm{T}} \qquad (2.17)$$

for $j = 1, \ldots, q$ and $k \geqslant j$, where $[\hat{D}_i(\zeta^{(t)})]_j$ is the jth row of

$$\begin{aligned} \hat{D}_i(\zeta^{(t)}) &= E\{c_i(a)\beta^{\mathrm{T}} | Y_{\mathrm{obs}}, \zeta^{(t)}\} \\ &= c_i(a, \theta^{(t)})\hat{\beta}^{\mathrm{T}}(\zeta^{(t)}) - \tilde{U}(2-a)\Delta^{\mathrm{T}} Z_i W_i(\zeta^{(t)}) X_i^{\mathrm{T}} \left\{\sum_{i=1}^{m} X_i W_i(\zeta^{(t)}) X_i^{\mathrm{T}}\right\}^{-1}. \end{aligned}$$

Alternatively, $\sigma^{2(t+1)}$ and $\tilde{\beta}^{(t+1)}$ can be computed via the SWEEP operator by using equations (2.15), (2.17) and

$$\begin{aligned} E\{(y_i - X_i^{\mathrm{T}}\beta)^{\mathrm{T}} R_i^{-1}(y_i - X_i^{\mathrm{T}}\beta) | Y_{\mathrm{obs}}, \zeta^{(t)}\} &= (y_i - X_i^{\mathrm{T}}\hat{\beta}(\zeta^{(t)}))^{\mathrm{T}} R_i^{-1}(y_i - X_i^{\mathrm{T}}\hat{\beta}(\zeta^{(t)})) \\ &\quad + \mathrm{tr}\left[X_i R_i^{-1} X_i^{\mathrm{T}} \left\{\sum_{i=1}^{m} X_i W_i(\zeta^{(t)}) X_i^{\mathrm{T}}\right\}^{-1}\right] \end{aligned}$$

to calculate the input for the SWEEP operator.

3. Simulation studies

3.1. Variance component models

Two sets of variance component simulations were conducted, one with data similar to a typical repeated measure analysis (i.e. n_i small and m large) and one similar to a common unbalanced analysis-of-variance analysis (i.e. n_i large and unequal, and m small). In the repeated measures simulation data were generated from the model

$$y_i = X^{\mathrm{T}}\beta + Z^{\mathrm{T}} b_i + e_i, \qquad i = 1, \ldots, m, \qquad (3.1)$$

where y_i is 2×1, $X = Z = (1, 1)$, $\beta = 1$, $b_i \sim N(0, 9)$ and $e_i \sim N_2(0, \sigma^2 I_2)$ with b_i and e_i independent.

As will be discussed in Section 4, the relative efficiency of the algorithms depends on the relative sizes of the variance of $Z^{\mathrm{T}} b$ and the residual variance σ^2. The simulation was therefore repeated with $\sigma^2 = 0.5$, 1, 4, 9, 16, 36, 49, 64, 81. For each of these values, we generated $m = 100$ observations from model (3.1). The starting values $\beta^{(0)}$ and $\sigma^{2(0)}$ were obtained by fitting model (3.1) ignoring the variance components, and $T^{(0)}$ was set to 1. We ran the standard ECME algorithm along with the $\mathrm{ECME}_{(1)}$ algorithm, i.e. the alternative algorithm with $Y_{\mathrm{aug}}(1)$ (i.e. with $a = 1$), and recorded t_{std} and t_{alt}—the time required (in seconds) by the standard and alternative algorithms respectively before the convergence criterion $L(\theta^{(t)} | Y_{\mathrm{obs}}) - L(\theta^{(t-1)} | Y_{\mathrm{obs}}) < 10^{-7}$ was reached. Of course the computation time depends on the machine used but comparisons of algorithms should be similar regardless of the machine. We used a Sun Sparc 4 computer and computed t_{std} and t_{alt} by multiplying the number of iterations required by the average central processor unit time per iteration for each algorithm (averaged over the total number of iterations cumulated over all the repetitions). The simulation was repeated 200 times and the results appear in Fig. 1, which displays a sequence of plots that highlight the computational savings that the $\mathrm{ECME}_{(1)}$ algorithm offers over the standard algorithm, especially when σ^{2*} is large relative to T^*. Fig. 1(a) displays the efficiency of the standard algorithm measured in $\log_{10}$(seconds) as a function of a measure of the overall coefficient of determination,

$$D^* = \frac{\sum_{i=1}^{m} \mathrm{tr}(Z_i^{\mathrm{T}} T^* Z_i)/m}{\sigma^{2*} + \sum_{i=1}^{m} \mathrm{tr}(Z_i^{\mathrm{T}} T^* Z_i)/m}.$$

See Section 4 for the theoretical result on the relationship between the rate of convergence and D^*; Foulley and Quass (1995) chose to plot their simulation results against a similar quantity, and their choice seems to be based on empirical findings.

It is clear from Fig. 1(a) that the standard algorithm becomes very slow when D^* is close to 0. Also plotted (as plus signs) is the $\log_{10}$(seconds) required by the commercially available `xtreg` routine in STATA for a single data set selected from the data sets corresponding to each of the values of σ^2 in the simulation configuration. Fig. 1(b) displays the time required (in $\log_{10}$(seconds)) by the $\mathrm{ECME}_{(1)}$ algorithm. This algorithm performs very well unless D^* is very small or very large. In particular it performs very well relative to the `xtreg` routine, which only allows for a single random effect and requires all elements of Z to be 1. Notice that `xtreg` requires between 4 and 7 s (median time 5 s) whereas $\mathrm{ECME}_{(1)}$ required only 0.08–2.90 s (median time 0.15 s). Of course, it is difficult to make a fair comparison of computational efficiency, since the computer time required is a function of the actual coding, the programming language and the machine. None-the-less, it is clear that `xtreg` (and the S-PLUS `lme` routine, as discussed in the next section) does not dominate $\mathrm{ECME}_{(1)}$, even in terms of computational time. Fig. 1(c) compares the standard algorithm with the $\mathrm{ECME}_{(1)}$ algorithm. The smaller D^*, the greater is the computational advantage of the new data augmentation scheme. When D^* is very large, the standard algorithm can be as much as 10 times faster. But, when D^* is less than about $\frac{2}{3}$, the $\mathrm{ECME}_{(1)}$ algorithm is preferable and can be several hundred times faster when D^* is close to 0, and is often 10–25 times faster.

The second set of variance component simulations were also under model (3.1), but only 10 groups were generated (i.e. $m = 10$), five of size 3 (i.e. $n_i = 3$) and five of size 7 (i.e. $n_i = 7$).

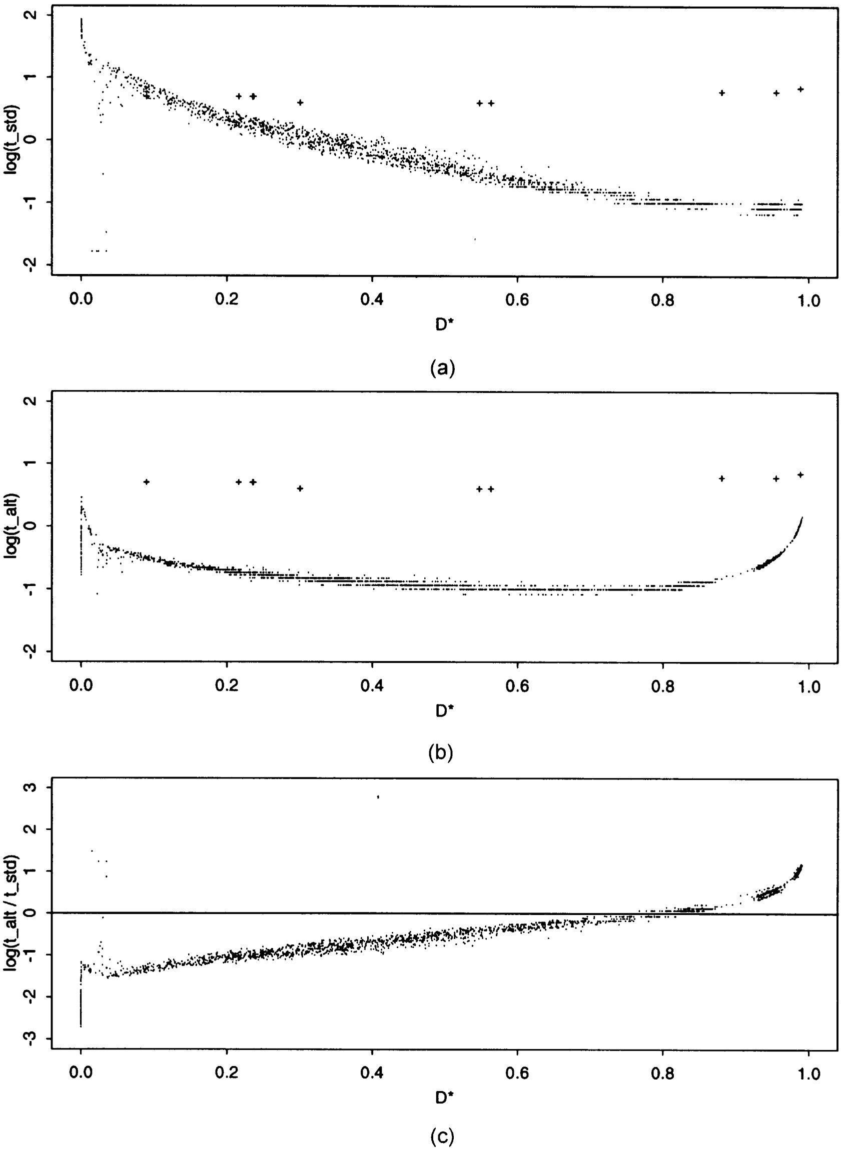

Fig. 1. Time required by the ECME algorithm as a function of the coefficient of determination D^*: (a) time (in $\log_{10}$(seconds)) required by the standard algorithm; (b) time (in $\log_{10}$(seconds)) required by the ECME$_{(1)}$ algorithm; (c) logarithm of the relative time (the ECME$_{(1)}$ algorithm performs better when D^* is smaller than about 2/3); $+$, time required by the `xtreg` routine in STATA for 10 randomly selected data sets

The simulation used the same starting values, convergence criterion and values of the parameters, and the results appear in Fig. 2, which plots the log-ratio of the computation time required by the ECME$_{(1)}$ algorithm and the standard algorithm against D^*. Again we see that the smaller D^* is the better ECME$_{(1)}$ performs relative to the standard algorithm, with ECME$_{(1)}$ preferable when $D^* < \frac{2}{3}$ (approximately), which is in good agreement with the theory given in Section 4.

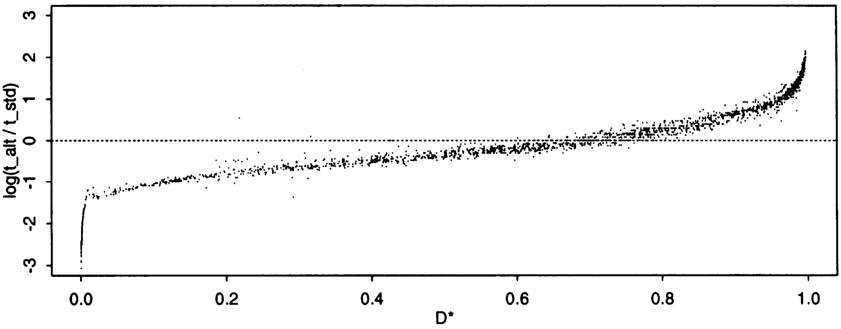

Fig. 2. Relative performance of the ECME$_{(1)}$ algorithm and the standard algorithm as a function of the coefficient of determination D^* for unbalanced data generated in the second simulation in Section 3.1

3.2. Mixed effects models

The second pair of simulation studies was similar to the first except that, to look at the more general mixed effects model, the components of Z_i were independently generated from the $N(0, 1)$ distribution and multivariate random effects were considered. In the first set of simulations, data were generated from the model

$$y_i = \beta_1 + x_i\beta_2 + Z_i^{\mathrm{T}} b_i + e_i, \qquad i = 1, \ldots, m, \tag{3.2}$$

where y_i and x_i are scalar, Z_i is 2×1, $\beta_1 = \beta_2 = 1$, $x_i = i$,

$$b_i \sim N_2\left\{0, \begin{pmatrix} 9 & 0 \\ 0 & 4 \end{pmatrix}\right\}$$

and $e_i \sim N(0, \sigma^2)$, with b_i and e_i independent. The second set used

$$y_i = X^{\mathrm{T}}\beta + Z_i^{\mathrm{T}} b_i + e_i, \qquad i = 1, \ldots, m, \tag{3.3}$$

where y_i is 2×1, $X = (1, 1)$, $\beta = 1$, Z_i is 2×2,

$$b_i \sim N_2\left\{0, \begin{pmatrix} 9 & 0 \\ 0 & 4 \end{pmatrix}\right\}$$

and $e_i \sim N_2(0, \sigma^2 I_2)$ with b_i and e_i independent. For each simulation, a data set with $m = 100$ was generated for each value of σ^2 (0.25, 1, 4, 9, 16, 25, 36, 49, 64, 81) and the same convergence criterion and starting values were used, except

$$T^{(0)} = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix}.$$

The first simulation set was repeated 200 times and the second set 100 times, and the results are summarized in Figs 3 and 4 respectively. Again, we want to compare the efficiency of the standard ECME algorithm with the ECME$_{(1,1)}$ algorithm (i.e. the new algorithm with $a = (1, 1)$) as a function of the coefficient of determination, and we plotted performance against $\mathrm{logit}_{10}(D^*)$ — the logit scale was used to highlight the comparisons when D^* is small or larger. For both models, the standard algorithm performs best when the residual variance is somewhat smaller than the average variance of $Z_i^{\mathrm{T}} b_i$ (i.e. when $\mathrm{logit}_{10}(D^*) \approx 1$) and the ECME$_{(1,1)}$ algorithm does very well when the residual variance is moderate to large relative to the average variance of $Z_i^{\mathrm{T}} b_i$. When the residual variance was small in the univariate response

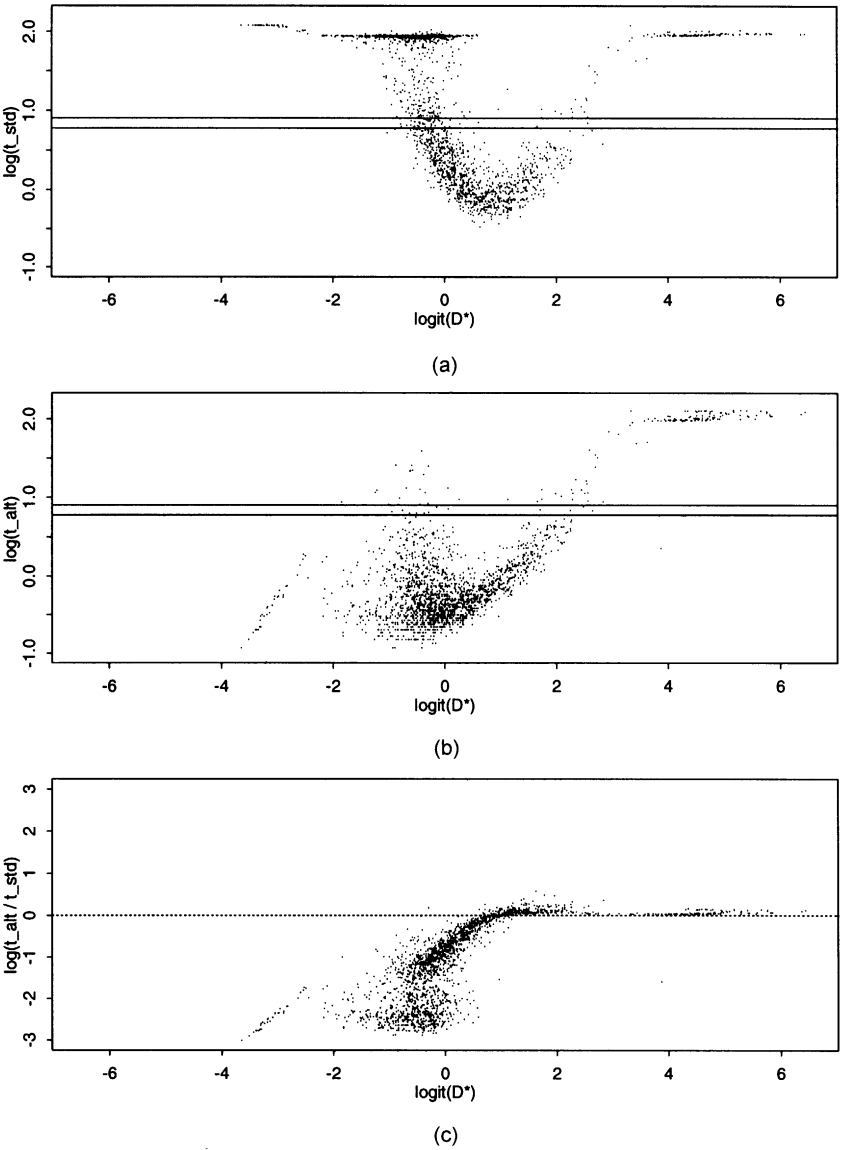

Fig. 3. Time required by the ECME algorithm as a function of $\text{logit}_{10}(D^*)$: (a) time required by the standard algorithm; (b) time required by the $\text{ECME}_{(1,1)}$ algorithm; (c) relative time; the standard algorithm performs better when the residual variance is small, and the $\text{ECME}_{(1,1)}$ algorithm performs better when the variance of the random effects is small; the parallel lines represent the range of time required by the `lme` routine in S-PLUS for 10 representative data sets; all times are reported in $\log_{10}$(seconds)

simulation, both algorithms are very slow as opposed to the variance component simulation in which both algorithms performed relatively well. Figs 3(c) and 4(c) indicate that, when D^* is greater than about 0.9, the standard algorithm tends to outperform the $\text{ECME}_{(1,1)}$ algorithm slightly (as much as 10 times faster). However, when the average variance of the random effects does not dominate the residual variance, the $\text{ECME}_{(1,1)}$ algorithm is clearly superior (as much as 1034 times faster).

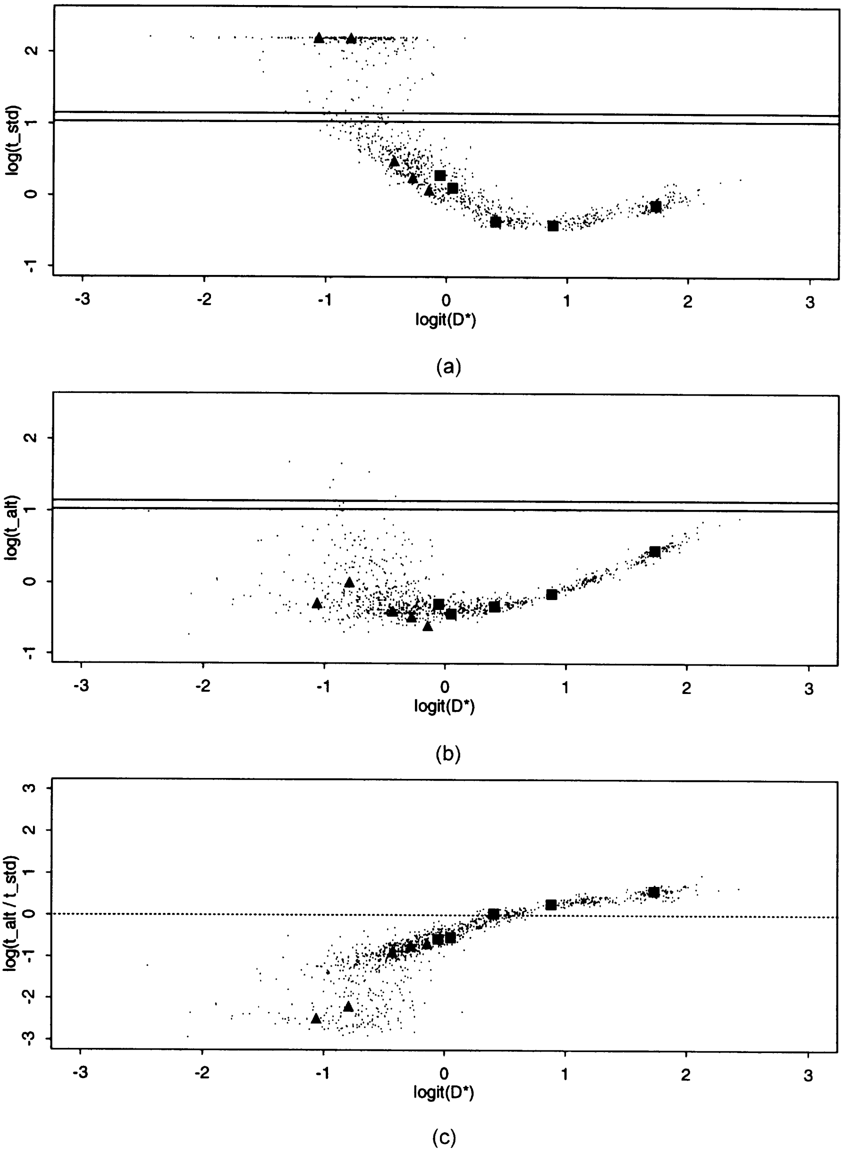

Fig. 4. Similar to Fig. 3 except that model (3.3) was used instead of model (3.2); ■, ▲, 10 arbitrarily selected data sets fitted using the lme routine for comparison; the vertical values of these symbols refer to the time (or relative time) for the ECME algorithm when fitting the corresponding data set; the range of time required by S-PLUS is within the parallel lines; for the data sets represented by triangles, S-PLUS converged to a point that was *lower* on the likelihood surface than did the ECME algorithm; all times are reported in $\log_{10}$(seconds)

The two lines in Figs 3(a) and 4(a) give the range of computation time required by the commercially available routine lme in S-PLUS for 10 data sets covering the range of the simulation. The $\mathrm{ECME}_{(1,1)}$ algorithm generally compares quite well with the lme routine and exhibits superior stability properties. For example, of the 10 data sets generated in the multivariate response simulation which were also fitted by lme (represented by squares and triangles in Fig. 4), the lme and ECME algorithms converged to different limits five times

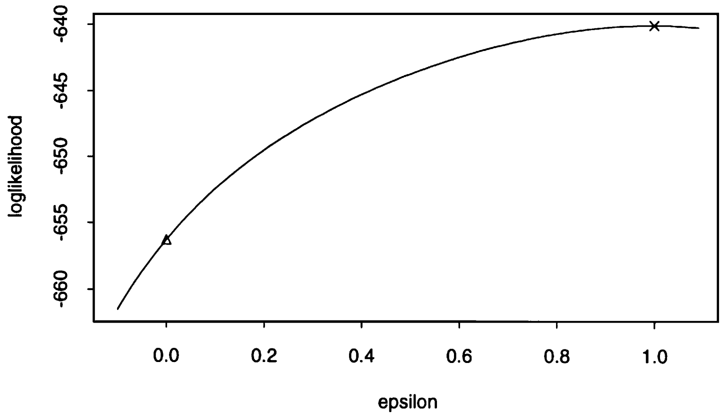

Fig. 5. Comparing the convergence of the ECME algorithm with the S-PLUS implementation: cross-section in the log-likelihood surface for the data set represented by the leftmost triangle in Fig. 4 (△, point of convergence of the S-PLUS implementation; X, point of convergence of the ECME algorithm; △ is clearly not even at a local mode)

(represented by triangles in Fig. 4). In all five cases the ECME algorithm converged to a point that is higher on the log-likelihood surface. One of these is represented in Fig. 5 which illustrates a cross-section of the log-likelihood surface (computed with our C code which agreed with the lme output) along the line $(1-\epsilon)\theta^*_{\mathrm{S}} + \epsilon\theta^*_{\mathrm{ECME}}$, where θ^*_{S} is the point of convergence of S-PLUS, θ^*_{ECME} is the point of convergence of ECME and ϵ ranges from 0 to 1. The S-PLUS routine does not even converge to a local mode. For this data set, the ECME algorithm converged to θ^*_{ECME} even when θ^*_{S} was used as the starting value. This is yet another illustration of the advantage of the stable convergence properties of the EM-type algorithms. Although these properties can come at the cost of slow convergence, this is a small price to pay for some assurance of the properties of the point of convergence. The central theme of efficient data augmentation is to maintain these properties while reducing the computational time.

The 10 data sets fitted with lme are available on request. We have used the same lme implementation for all the 10 data sets and it provided the same answer as our ECME implementation for five sets. We make no claim of any kind about the general reliability of the lme routine. We are simply reporting what we have encountered in our implementation of it, which, to our best knowledge, was in accordance with the instructions in the S-PLUS manual.

3.3. *An adaptive algorithm*

Although the relative gain of the standard algorithm over the $\mathrm{ECME}_{(1,1)}$ algorithm is small when the residual variance is very small, cutting the computational time even in half can be significant since both algorithms can be so slow in this case (e.g. Fig. 3). To take advantage of the standard algorithm when it is more efficient, a preliminary approximation of θ^* can be used to decide between the two algorithms. To invesigate this, we repeated the univariate response random effects simulation (with new random seeds) with an adaptive algorithm, which first runs the $\mathrm{ECME}_{(1,1)}$ algorithm for 20 iterations and then switches to the standard algorithm if

$$2q[\sigma^2]^{(20)} \leqslant \frac{1}{m}\sum_{i=1}^{m} \mathrm{tr}(Z_i^{\mathrm{T}} T^{(20)} Z_i). \qquad (3.4)$$

This criterion is based on an extension of the results of Section 4 to the case with $q > 1$ (see van Dyk (1995)).

Fig. 6 plots $\log_{10}(N_{\mathrm{adp}}/N_{\mathrm{std}})$ against $\mathrm{logit}_{10}(D^*)$, where N_{adp} and N_{std} are the number of iterations required by the adaptive and the standard algorithms respectively. Displaying the number of iterations required instead of time makes it a little easier to detect whether a switch has occurred—comparisons using actual time do not alter the overall pattern of the plots (for example, see Fig. 7). As Fig. 6 indicates, the adaptive algorithm almost always switched to the standard algorithm when it was beneficial to do so. (Interestingly, the switched algorithm was often slightly faster than the pure standard algorithm.) Since this adaptive algorithm is easy to implement and generally performs well against both the ECME$_{(1,1)}$ and the standard algorithm, we recommend its use in place of either the standard or ECME$_{(1,1)}$ algorithms. We also expect that a switching criterion that is more effective than condition (3.4) can be found.

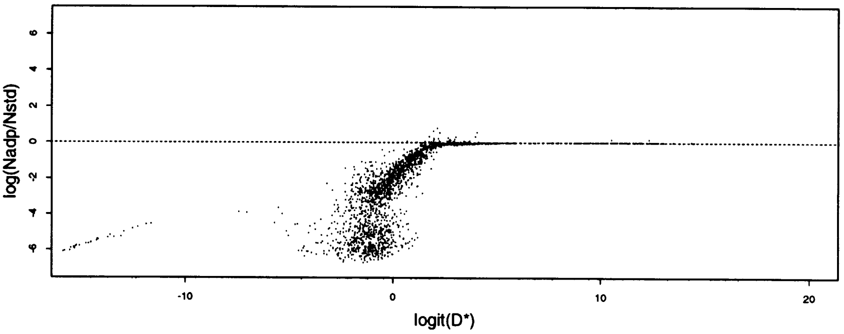

Fig. 6. Relative number of iterations required by the adaptive and standard algorithm ($\log_{10}$-scale): after 20 iterations of the ECME$_{(1,1)}$ algorithm, the current approximation $\theta^{(20)}$ was used to determine which algorithm should be used; if $4[\sigma^2]^{(20)} < (1/m)\,\Sigma_i\, Z_i^{\mathrm{T}} T^{(20)} Z_i$, the standard algorithm was used until convergence; otherwise we continued with the ECME$_{(1,1)}$ algorithm until convergence; this procedure almost always resulted in an algorithm that was faster than the standard algorithm

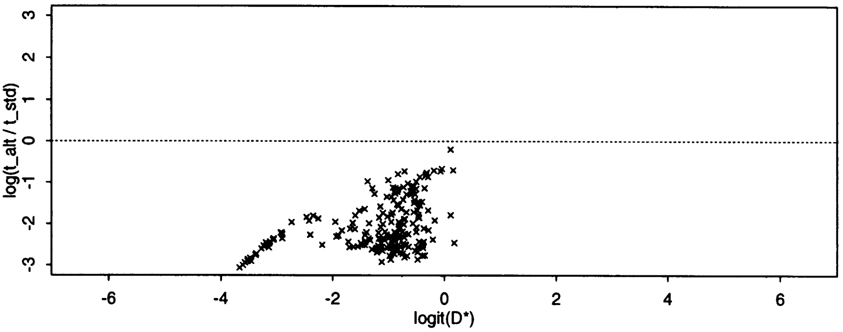

Fig. 7. Computational gain when the T^* is small: when T^* is small, the reparameterization of T induced by the new data augmentation, in effect, keeps T^* from the boundary of the parameter space and thus helps to increase the computational efficiency of the ECME algorithm

3.4. The EM boundary problem

It is well known that the EM-type algorithm can be very slow to converge when the maximum likelihood estimate is near the boundary of the parameter space (e.g. with the image reconstruction problem; see Meng and van Dyk (1997), section 3.5, for discussion). This observation may lend partial insight into the computational gains that our new data augmentation scheme avails. Consider the case where the random effect is univariate. By moving the estimation of T into the mean structure at each M-step, the parameter space for the variance, $[0, \infty)$, is transformed to the parameter space for a regression coefficient, $(-\infty, \infty)$. Thus, small values of the variance T^* which are near the boundary of $[0, \infty)$ are transformed to points that are far from the boundary of $(-\infty, \infty)$. To illustrate this, we conducted another simulation with data generated from model (3.2), but this time with

$$b_i \sim N_2\left\{0, \begin{pmatrix} 0.01 & 0 \\ 0 & 0.02 \end{pmatrix}\right\}$$

and $\sigma^2 = 4$. The results of the simulation appear in Fig. 7 and demonstrate the magnitude of the computational gain of the new data augmentation when T^* is relatively small, with an average reduction in computational time of a factor greater than 65. Thus avoiding the boundary problem seems to be a reasonable interpretation of why the relative size of the variance of $Z^{\mathrm{T}}b$ compared with the residual variance is the driving force behind the speed of convergence. Consequently, in the absence of other information, we recommend the use of the ECME algorithm that transfers all of T into the mean structure (i.e. using $a = (1, 1, \ldots, 1)^{\mathrm{T}}$), which avoids the potential boundary problem for any particular component.

The very slow convergence of EM-type algorithms sometimes can be taken as an indication of problematic aspects of the underlying model specification. In Fig. 3, data sets in which the maximum likelihood estimate of the correlation of the two random effects was close to 1 in absolute value (and thus a single random effect may be sufficient) were among those for which the new algorithm offered the most improvement. Fig. 3 indicates, however, that slow convergence of the standard algorithm can be attributed to at least two aspects of the model (i.e. either σ^{2*} is small or T^* is almost singular). A quick computation of the correct maximum likelihood estimate (in contrast with a fast but unstable algorithm which may converge to a point that has no statistical significance) retains this information and facilitates a diagnosis of model misspecification.

4. Theoretical derivations

4.1. A useful theoretical simplification

The theory behind choosing an efficient augmentation scheme for mixed effects models is considerably more complicated than for the t-models presented in Meng and van Dyk (1997) (but the complication is for those who design the algorithms, not for general users). The main difficulty is that the expected augmented data information matrix $I_{\mathrm{aug}}(a)$ is generally of large dimension and has a complicated structure. Specifically, the dimension of the parameter $\theta = (\beta, \Delta, U, \sigma^2)$ is $p + q(q+1)/2 + 1$, and $I_{\mathrm{aug}}(a)$ consists of the submatrices

$$I_{\mathrm{aug}}(a) = \begin{pmatrix} I_{\beta\beta}(a) & I_{\beta\Delta}(a) & I_{\beta U}(a) & I_{\beta\sigma^2}(a) \\ I^{\mathrm{T}}_{\beta\Delta}(a) & I_{\Delta\Delta}(a) & I_{\Delta U}(a) & I_{\Delta\sigma^2}(a) \\ I^{\mathrm{T}}_{\beta U}(a) & I^{\mathrm{T}}_{\Delta U}(a) & I_{UU}(a) & I_{U\sigma^2}(a) \\ I^{\mathrm{T}}_{\beta\sigma^2}(a) & I^{\mathrm{T}}_{\Delta\sigma^2}(a) & I^{\mathrm{T}}_{U\sigma^2}(a) & I_{\sigma^2\sigma^2}(a) \end{pmatrix}.$$

It is not difficult to show, by differentiating the expected augmented data log-likelihood (without loss of generality here we assume that $R_i = I_{n_i}$),

$$Q(\beta, \Delta, U, \sigma^2|\theta^{(t)}) = -\frac{n}{2}\log(\sigma^2) - \frac{m}{2}\sum_{j=1}^{q}(1-a_j)\log(u_j^2) - \frac{1}{2}\sum_{i=1}^{m} E[c_i^{\mathrm{T}}(a)\tilde{U}\{2(a-1)\}c_i(a)|\theta^{(t)}, Y_{\mathrm{obs}}] - \frac{1}{2\sigma^2}\sum_{i=1}^{m} E\{(y_i - X_i^{\mathrm{T}}\beta - Z_i^{\mathrm{T}}\Delta\tilde{U}(a)c_i(a))^{\mathrm{T}}(y_i - X_i^{\mathrm{T}}\beta - Z_i^{\mathrm{T}}\Delta\tilde{U}(a)c_i(a))|\theta^{(t)}, Y_{\mathrm{obs}}\}, \tag{4.1}$$

that (when evaluated at $\theta = \theta^*$) $I_{\beta\sigma^2}(a) = 0$, $I_{\Delta\sigma^2}(a) = 0$, $I_{U\sigma^2}(a) = 0$ and $I_{\beta\beta}(a)$ and $I_{\sigma^2\sigma^2}(a)$ do not depend on a. Furthermore, when $E(y_i|X_i, Z_i, \theta) = X_i^{\mathrm{T}}\beta$, i.e. when the mean structure of the posited model is correctly specified, $\lim_{n\to\infty}\{I_{\beta\Delta}(a)/n\} = 0$ and $\lim_{n\to\infty}\{I_{\beta U}(a)/n\} = 0$. Thus, as long as n is not too small, the only part of $I_{\mathrm{aug}}(a)$ that can change substantially with a is the $[q(q+1)/2] \times [q(q+1)/2]$ submatrix

$$\widetilde{I_{\mathrm{aug}}}(a) = \begin{pmatrix} I_{\Delta\Delta}(a) & I_{\Delta U}(a) \\ I_{\Delta U}^{\mathrm{T}}(a) & I_{UU}(a) \end{pmatrix}. \tag{4.2}$$

In fact, even when $I_{\beta\Delta}(a)$ or $I_{\beta U}(a)$ are non-zero, we expect that they have less effect on the smallest eigenvalue of the speed matrix relative to the effect of $\widetilde{I_{\mathrm{aug}}}(a)$ because the positiveness of $I_{\mathrm{aug}}(a)$ requires that off-diagonal blocks be dominated by the diagonal blocks. Furthermore, for the ECME implementation described in Section 2.2, $\widetilde{I_{\mathrm{aug}}}(a)$ plays a more direct role in the corresponding rate of convergence (see theorem 4 of Meng and van Dyk (1997) for a derivation). We thus focus on equation (4.2) when we search for optimal, or good, values of a.

4.2. *Derivation for a scalar random effect*

We shall apply Meng and van Dyk's (1997) theorem 1 which requires us to order (in the positive semidefinite ordering sense) $I_{\mathrm{aug}}(a)$, which, as we have seen, is approximately equivalent to ordering $\widetilde{I_{\mathrm{aug}}}(a)$. For simplicity, we consider the case of one random effect (i.e. $q = 1$); for more than one random effect, see van Dyk (1995) or Meng and van Dyk (1995). When $q = 1$, $\widetilde{I_{\mathrm{aug}}}(a)$ is a scalar and equation (4.1) reduces to

$$Q(\beta, u^2, \sigma^2|\theta^{(t)}) = -\frac{n}{2}\log(\sigma^2) - \frac{m}{2}(1-a)\log(u^2) - \frac{1}{2}\sum_{i=1}^{m}\frac{\hat{B}_i(a, \theta^{(t)})}{u^{2(1-a)}} - \frac{1}{2\sigma^2}\sum_{i=1}^{m}\{(y_i - X_i^{\mathrm{T}}\beta)^{\mathrm{T}}(y_i - X_i^{\mathrm{T}}\beta) - 2(y_i - X_i^{\mathrm{T}}\beta)^{\mathrm{T}}Z_i^{\mathrm{T}}\hat{c}_i(a, \theta^{(t)})u^a + Z_iZ_i^{\mathrm{T}}\tilde{B}_i(a, \theta^{(t)})u^{2a}\}. \tag{4.3}$$

Differentiating equation (4.3) twice with respect to u^2 and evaluating at $\theta = \theta^*$, we obtain (details are given in van Dyk (1995))

$$\widetilde{I_{\mathrm{aug}}}(a) = \frac{1}{2u^{4*}}\left[a^2\left\{\frac{1}{2\sigma^{2*}}\sum_{i=1}^{m}\mathrm{tr}(Z_i^{\mathrm{T}}T_i^*Z_i)\right\} + (1-a)^2 m\right], \tag{4.4}$$

where $T_i^* = E(b_i^2|Y_{\mathrm{obs}}, \theta^*)$. It is easy to see that $\widetilde{I_{\mathrm{aug}}}(a)$ is minimized as a function of a for

$$a_o = \left\{ \frac{\sum_{i=1}^{m} \mathrm{tr}(Z_i^{\mathrm{T}} T_i^* Z_i)/m}{2\sigma^{2*}} + 1 \right\}^{-1} = \frac{2(1-\tilde{D}^*)}{2-\tilde{D}^*}, \tag{4.5}$$

where

$$\tilde{D}^* = \frac{\sum_{i=1}^{m} \mathrm{tr}(Z_i^{\mathrm{T}} T_i^* Z_i)/m}{\sigma^{2*} + \sum_{i=1}^{m} \mathrm{tr}(Z_i^{\mathrm{T}} T_i^* Z_i)/m} \tag{4.6}$$

is a measure of the overall coefficient of determination. Note that $\tilde{D}^*$ is slightly different from the D^* that we used in the plots; the latter replaces T_i^* by T^* and thus is directly calculable from the maximum likelihood estimate of T (when Z_i does not depend on i, as in common variance component models, $D^* = \tilde{D}^*$). Unfortunately, implementing the ECME algorithm with augmented data $Y_{\mathrm{aug}}(a_o)$ results in a rather complicated M-step and thus does not satisfy our objective of using simple algorithms. We thus confine our attention to algorithms which result from $a = 0$ or $a = 1$ (or, more generally, for q random effects $a \in \{0, 1\}^q$); equation (4.5) suggested that we do not need to consider a outside [0, 1]. In this class of algorithms, equation (4.4) is minimized by

$$a_{\mathrm{opt}} = \begin{cases} 0 & \text{if } \dfrac{1}{m} \sum_{i=1}^{m} \mathrm{tr}(Z_i^{\mathrm{T}} T_i^* Z_i) > 2\sigma^{2*}, \\ 1 & \text{otherwise,} \end{cases} = \begin{cases} 0 & \text{if } \tilde{D}^* \geqslant \frac{2}{3}, \\ 1 & \text{otherwise,} \end{cases} \tag{4.7}$$

i.e. the (finite sample) augmented information $\widetilde{I_{\mathrm{aug}}}(a)$ is smaller for the new augmentation (i.e. $a = 1$) than for the standard augmentation (i.e. $a = 0$) if and only if $D^* < \frac{2}{3}$. In other words, we take $a_{\mathrm{opt}} = 1$ if $a_o > \frac{1}{2}$, $a_{\mathrm{opt}} = 0$ if $a_o < \frac{1}{2}$ and $a_{\mathrm{opt}} = 0$ or $a_{\mathrm{opt}} = 1$ if $a_o = \frac{1}{2}$.

The obvious difficulty with the condition in expression (4.7) is that it depends on the parameter values (unlike the *t*-model presented in Meng and van Dyk (1997)). Happily, however, our limited empirical studies suggest that the standard algorithm, relatively speaking, is only slightly more efficient when σ^{2*} is small and that using $a = (1, \ldots, 1)^{\mathrm{T}}$ will generally lead to computational advantages and occasionally will lead to a slight disadvantage. Moreover, the adaptive algorithm discussed in Section 3.3 can help to eliminate even this slight disadvantage.

5. A concluding remark

We conclude by noting that the new data augmentation scheme described here is also useful for implementing the Gibbs sampler. The EM algorithms described here can readily be adapted to find posterior modes, which can be used to construct starting values for the Gibbs sampler (e.g. Gelman *et al.* (1995), chapters 9–11). Moreover, efficient data augmentation schemes are very useful in the Gibbs sampler itself where fast convergence is especially important since often in practice a user must stop running a sampler simply because he cannot afford to run it longer. Preliminary investigations indicate that the data augmentation scheme introduced here can substantially reduce autocorrelation in the Gibbs sampler, when the coefficient of determination is not too large. This may be useful in conjunction with the recentring parameterization proposed by Gelfand *et al.* (1995), who argued that such a parameterization works well when the coefficient of determination is large. For a more detailed discussion of this and related topics, including possibly even better data augmentation schemes for both the EM algorithm and the Gibbs sampler, see Meng and van Dyk (1997), section 4, and the accompanying discussions and the rejoinder.

Acknowledgements

The research was supported in part by National Science Foundation (NSF) grants DMS 92-04504, DMS 95-05043, DMS 96-26691 and DMS 97-05156, in part by National Security Agency grant MDA 904-9610007 and in part by the US Census Bureau through a contract with the National Opinion Research Center at the University of Chicago. The manuscript was prepared using computer facilities supported in part by several NSF grants and by the University of Chicago Block Fund. We thank Y. Amemiya, A. Gelman, D. Harville and the reviewers for helpful suggestions and comments. In particular, we thank a referee for bringing to our attention the work by Foulley and Quass (1995).

References

Anderson, D. A. and Aitkin, M. (1985) Variance component models with binary response: interviewer variability. *J. R. Statist. Soc.* B, **47**, 203–210.

Beaton, A. E. (1964) The use of special matrix operations in statistical calculus. *Research Bulletin RB-64-51.* Education Testing Service, Princeton.

Callanan, T. P. and Harville, D. A. (1991) Some new algorithms for computing restricted maximum likelihood estimates of variance components. *J. Statist. Computn Simuln*, **38**, 239–259.

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

Dempster, A. P., Selwyn, M. R., Patel, C. M. and Roth, A. J. (1984) Statistical and computational aspects of mixed model analysis. *Appl. Statist.*, **33**, 203–214.

van Dyk, D. A. (1995) Construction, implementation, and theory of algorithms based on data augmentation and model reduction. *PhD Thesis.* Department of Statistics, University of Chicago, Chicago.

Foulley, J. L. and Quass, R. L. (1995) Heterogeneous variance in Gaussian linear mixed models. *Genet. Selectn Evoln*, **27**, 211–228.

Gelfand, A. E., Sahu, S. K. and Carlin, B. P. (1995) Efficient parameterizations for normal linear mixed models. *Biometrika*, **82**, 479–488.

Gelman, A., Carlin, J., Stern, H. and Rubin, D. B. (1995) *Bayesian Data Analysis.* London: Chapman and Hall.

Groeneveld, E. (1994) A reparameterization to improve numerical optimization in multivariate REML (co)variance component estimation. *Genet. Selectn Evoln*, **26**, 537–545.

Harville, D. A. (1977) Maximum likelihood approaches to variance component estimation and to related problems. *J. Am. Statist. Ass.*, **72**, 320–340.

Horn, R. A. and Johnson, C. R. (1985) *Matrix Analysis.* New York: Cambridge University Press.

Laird, N. M. (1982) Computation of variance components using the E-M algorithm. *J. Statist. Computn Simuln*, **14**, 295–303.

Laird, N., Lange, N. and Stram, D. (1987) Maximizing likelihood computations with repeated measures: application of the EM algorithm. *J. Am. Statist. Ass.*, **82**, 97–105.

Laird, N. M. and Ware, J. H. (1982) Random effects models for longitudinal data. *Biometrics*, **38**, 967–974.

Lindstrom, M. J. and Bates, D. M. (1988) Newton–Raphson and EM algorithms for linear mixed-effects models for repeated-measure data. *J. Am. Statist. Ass.*, **83**, 1014–1022.

Little, R. J. A. and Rubin, D. B. (1987) *Statistical Analysis with Missing Data.* New York: Wiley.

Liu, C. and Rubin, D. B. (1994) The ECME algorithm: a simple extension of EM and ECM with fast monotone convergence. *Biometrika*, **81**, 633–648.

Meng, X.-L. and van Dyk, D. A. (1995) The EM algorithm—an old folk song sung to a fast new tune. *Technical Report 408.* Department of Statistics, University of Chicago, Chicago.

——— (1997) The EM algorithm—an old folk-song sung to a fast new tune (with discussion). *J. R. Statist. Soc.* B, **59**, 511–567.

Meng, X.-L. and Pedlow, S. (1992) EM: a bibliographic review with missing articles. *Proc. Statist. Comput. Sect. Am. Statist. Ass.*, 24–27.

Thompson, R. (1995) Comment on Heterogeneous variance in Gaussian linear mixed models by JL Foulley and RL Quass. *Genet. Selectn Evoln*, **27**, 226–227.

Thompson, R. and Meyer, K. (1986) Estimation of variance components: what is missing in the EM algorithm? *J. Statist. Computn Simuln*, **24**, 215–230.

Wu, C. F. J. (1983) On the convergence properties of the EM algorithm. *Ann. Statist.*, **11**, 95–103.