On the Orderings and Groupings of Conditional Maximizations Within ECM-Type Algorithms

David A. VAN DYK and Xiao-Li MENG

The ECM and ECME algorithms are generalizations of the EM algorithm in which the maximization (M) step is replaced by several conditional maximization (CM) steps. The order that the CM-steps are performed is trivial to change and generally affects how fast the algorithm converges. Moreover, the same order of CM-steps need not be used at each iteration and in some applications it is feasible to group two or more CM-steps into one larger CM-step. These issues also arise when implementing the Gibbs sampler, and in this article we study them in the context of fitting log-linear and random-effects models with ECM-type algorithms. We find that some standard theoretical measures of the rate of convergence can be of little use in comparing the computational time required, and that common strategies such as using a random ordering may not provide the desired effects. We also develop two algorithms for fitting random-effects models to illustrate that with careful selection of CM-steps, ECM-type algorithms can be substantially faster than the standard EM algorithm.

Key Words: Contingency table; Convergence rate; Data augmentation; Gibbs sampler; EM algorithm; Incomplete data; IPF; Missing data; Model reduction; Random-effects models.

1. INTRODUCTION AND THEORETICAL BACKGROUND

1.1 CONDITIONAL MAXIMIZATIONS AND FLEXIBLE DATA AUGMENTATION

The EM algorithm (Dempster, Laird, and Rubin 1977) is a popular data-augmentation method for calculating maximum likelihood estimates (and posterior modes) in problems with incomplete data or problems that can be formulated as such (e.g., mixture models). The ECM algorithm (Meng and Rubin 1993) extends the EM algorithm by replacing the maximization (M) step of EM with several conditional maximization (CM) steps. The rationale behind this replacement is that for many common statistical models, direct maximization of the likelihood function is complicated even without missing data, but conditional maximizations of the likelihood function over constrained parameter spaces

David A. van Dyk is Assistant Professor, Department of Statistics, Harvard University, Cambridge, MA 02138; e-mail: vandyk@stat.harvard.edu. Xiao-Li Meng is Associate Professor, Department of Statistics, University of Chicago, Chicago, IL 60615.

Journal of Computational and Graphical Statistics, Volume 6, Number 2, Pages 202–223

can be much easier; this is in the spirit of the Gauss–Seidel method. Although this strategy of using conditional maximizations was motivated by the simplicity and stability (e.g., monotone convergence in likelihood) of the resulting algorithms, it turns out that it is also of fundamental importance in speeding up EM-type algorithms because it allows for much more flexible data-augmentation schemes than does the original EM algorithm (see Meng and van Dyk (in press) for a detailed discussion).

Specifically, let $L(\theta|Y_{\text{aug}})$ be the log-likelihood function based on the augmented data, Y_{aug}, where $\theta = (\theta_1, \ldots, \theta_d)$ is a d-dimensional model parameter with domain Θ. The objective here is to maximize the observed-data log-likelihood, $L(\theta|Y_{\text{obs}})$, where the observed data, Y_{obs}, is a function of Y_{aug}. Let $G = \{g_s(\theta), s = 1, \ldots, S\}$ be a set of $S \geq 1$ preselected (vector) constraint functions that are "space filling" (Meng and Rubin 1993), in the sense of allowing maximization over the full space Θ. Starting with an initial value $\theta^{(0)} \in \Theta$, the expectation (E) step of ECM is the same as in EM: find the conditional expectation of $L(\theta|Y_{\text{aug}})$ given $\theta = \theta^{(t)}$ and Y_{obs}

$$Q(\theta|\theta^{(t)}) = \mathrm{E}[L(\theta|Y_{\text{aug}})|\theta^{(t)}, Y_{\text{obs}}]. \tag{1.1}$$

The sth ($s = 1, \ldots, S$) CM-step of the $(t+1)$st iteration of ECM determines $\theta^{(t+s/S)}$ by maximizing $Q(\theta|\theta^{(t)})$ under the constraint $g_s(\theta) = g_s(\theta^{(t+(s-1)/S)})$; that is, by finding $\theta^{(t+s/S)}$ such that

$$Q(\theta^{(t+s/S)}|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}), \quad \text{for all} \quad \theta \in \{\theta \in \Theta : g_s(\theta) = g_s(\theta^{(t+(s-1)/S)})\}. \tag{1.2}$$

The output for the $(t+1)$st iteration is defined as $\theta^{(t+1)} \equiv \theta^{(t+S/S)}$, which can then be used to recompute (1.1) and the algorithm iterates until convergence. The original EM algorithm is a special case of ECM with $S = 1$ and $g_1(\theta)$ equal to a constant (i.e., no constraint).

The ECM algorithm thus generalizes EM by incorporating model reduction (i.e., breaking a big model into several smaller ones) into the M-step in order to regain the simplicity of EM in cases where the maximization of $Q(\theta|\theta^{(t)})$ over Θ is difficult. As expected, replacing the M-step by a sequence of CM-steps generally slows down convergence in terms of the rate of convergence (defined in Section 1.3; surprisingly, this is not universally true; see the counter-example provided by Meng 1994). The expectation/conditional maximization either (ECME) algorithm (Liu and Rubin 1994) uses a creative data-augmentation scheme to improve the speed of convergence. Liu and Rubin (1994) recognized that in some applications of the ECM algorithm the implementation of some CM-steps requires similar computations for maximizing the conditional observed-data likelihood and the conditional augmented-data likelihood, and thus, it can be computationally more efficient to directly maximize the former. Specifically, in ECME the first S_0 CM-steps are the same as in (1.2) but the rest of the CM-steps are replaced by finding $\theta^{(t+s/S)}$ such that

$$L(\theta^{(t+s/S)}|Y_{\text{obs}}) \geq L(\theta|Y_{\text{obs}}), \quad \text{for all} \quad \theta \in \{\theta \in \Theta : g_s(\theta) = g_s(\theta^{(t+(s-1)/S)})\}. \tag{1.3}$$

The idea of using different data-augmentation schemes at different CM-steps was also explored independently by Fessler and Hero (1994, 1995), who proposed a space-alternating generalized EM (SAGE) algorithm that allows Q to vary from CM-step to

CM-step. Combining the flexible model-reduction scheme of ECM and the flexible data-augmentation scheme of SAGE, Meng and van Dyk (in press) proposed a general framework called the alternating expectation/conditional maximization (AECM) algorithm for constructing simple, stable, and fast ECM-type algorithms for practical implementation. They also introduced the idea of a "working parameter" in order to facilitate the search for efficient data-augmentation schemes. The fast algorithms developed in this article for fitting (univariate response) random-effects models are built upon these ideas.

1.2 THE ISSUES OF ORDERING AND GROUPING

In both ECM and ECME (and more generally SAGE and AECM), the CM-steps maximize the objective function over a different subspace of Θ, so that the "space filling" condition is required to guarantee that the whole parameter space Θ will be searched after we perform all S CM-steps, but there is flexibility in the order we choose to perform them. On the other hand, the order of the CM-steps can sometimes significantly affect the number of iterations required for convergence. Taking an extreme example, when the ECM algorithm was used to fit a log-linear model to a certain sparse contingency table with partially classified counts, the ECM algorithm with one order of CM-steps required 10 times as many iterations (i.e., CPU time) as another order. A question of practical interest naturally arises: is there any way that we can take advantage of efficient orderings or at least avoid the bad orderings in cases where order has a substantial effect?

A second question of interest in practice involves groupings of CM-steps. In many problems the CM-steps are defined by breaking the parameter into several pieces, $\theta = (\vartheta_1, \ldots, \vartheta_S)$, and optimizing the objective function over each ϑ_s in sequence, conditional on the rest of θ (i.e., $g_s(\theta) = (\vartheta_1, \ldots, \vartheta_{s-1}, \vartheta_{s+1}, \ldots, \vartheta_S)$ for $s = 1, \ldots, S$). In such problems, it is sometimes possible to construct algorithms by grouping the components of θ in different ways. Although intuitively it seems that the bigger these groups are the more efficient the algorithms will be, this is not always true because we can sometimes use more efficient data-augmentation when the parameter is broken into smaller pieces (e.g., as with the ECME algorithm). Both the issue of ordering and the issue of grouping also arise naturally when implementing the Gibbs sampler, which can be viewed as a stochastic counterpart of ECM-type algorithms (see the discussion in Meng and Rubin 1992; Meng and van Dyk in press). Indeed, some of the strategies we investigate in this article were motivated by similar strategies for implementing the Gibbs sampler (e.g., Amit and Grenander 1991; Liu, Wong, and Kong 1994, 1995).

Our presentation is organized as follows. After we introduce the necessary background in Section 1.3, we will begin in Section 2 by using the common contingency table problem to illustrate the impact of changing orderings on the actual number of steps required for convergence. In Section 3 we investigate the effect of the ordering and grouping of CM-steps on the computation time required for convergence by several ECME algorithms used to fit random-effects models. We will also show that by careful construction of the CM-steps we can provide an implementation that is not only simple and stable but can also be much faster than the standard EM implementation for random-effects models. In Section 4 we provide concluding remarks.

1.3 SOME THEORETICAL BACKGROUND

Here we discuss the theoretical background of the ECM algorithm. Some of this discussion is also applicable to the more general AECM algorithm, but Theorem 1 only applies to ECM. Like any deterministic iterative algorithm, the ECM algorithm implicitly defines a mapping $M^{\text{ECM}} : \theta^{(t)} \rightarrow \theta^{(t+1)} = M^{\text{ECM}}(\theta^{(t)})$ from the parameter space Θ to itself. Let $\theta^\star$ be the limit of $\{\theta^{(t)} : t \geq 0\}$. Suppose that $M^{\text{ECM}}(\theta)$ is differentiable in a neighborhood of $\theta^\star$, then a Taylor's series argument shows that for large t, ECM can be approximated by a linear iteration with iteration matrix $DM^{\text{ECM}}(\theta^\star)$, the Jacobian of the mapping $M^{\text{ECM}}(\theta)$ evaluated at $\theta^\star$; thus $DM^{\text{ECM}}(\theta^\star)$ is called the (matrix) rate of convergence of ECM (e.g., Meng 1994).

The spectral radius of $DM^{\text{ECM}}(\theta^\star)$ is of particular interest because it is equal to the so called root convergence factor $\rho \equiv \limsup \sqrt[t]{||\theta^{(t)} - \theta^\star||}$ (e.g., see Ortega and Rheinboldt 1970, sec. 10.1.4). For computational purposes, we will use the empirical root convergence factor $\hat{\rho}_t = \left[\prod_{i=2}^{t} \hat{r}_i\right]^{1/(t-1)}$, where $\hat{r}_i = ||\theta^{(i)} - \theta^{(i-1)}||/||\theta^{(i-1)} - \theta^{(i-2)}||$, $i \geq 2$. It is worth noting that if DM^{ECM} has a spectral decomposition after a similarity transformation (as is always the case with the EM mapping (see Meng and Rubin 1994)), then $r = \lim_{t\rightarrow\infty} \hat{r}_t$ exists and is equal to the spectral radius of DM^{ECM}. Thus, in the EM literature r is used instead of ρ to assess the rate of convergence of the algorithm. This will not suffice in the ECM case as $\lim_{t\rightarrow\infty} \hat{r}_t$ may not exist and $\limsup_{t\rightarrow\infty} \hat{r}_t$ can be greater than the spectral radius; an example of this was given by van Dyk and Meng (1995).

In practice, one quantity of real interest is the actual number of iterations an algorithm requires for convergence. If $\hat{\rho}_t$ can be well approximated by ρ, then the number of iterations required for convergence, N, is related to ρ via

$$N \propto \frac{-1}{\log(\rho)}, \tag{1.4}$$

where the constant of proportionality depends only on the starting value and the convergence criterion. We therefore would like to study how ρ varies with different orderings of CM-steps. Let I_{obs} and I_{aug} be the observed and augmented information matrices:

$$I_{\text{obs}} = -\frac{\partial^2 L(\theta|Y_{\text{obs}})}{\partial\theta \cdot \partial\theta}\bigg|_{\theta=\theta^\star}, \qquad I_{\text{aug}} = \text{E}\left[-\frac{\partial^2 L(\theta|Y_{\text{aug}})}{\partial\theta \cdot \partial\theta}\bigg|Y_{\text{obs}}, \theta\right]_{\theta=\theta^\star}.$$

Meng (1994) showed that, suppressing the dependency on $\theta = \theta^\star$,

$$[I - DM^{\text{ECM}}] = [I - DM^{\text{EM}}][I - DM^{\text{CM}}], \tag{1.5}$$

where I is the identity matrix, $DM^{\text{EM}} = I - I_{\text{obs}} I_{\text{aug}}^{-1}$, and

$$DM^{\text{CM}} = I_{\text{aug}}^{\frac{1}{2}} \{A_1 \ldots A_S\} I_{\text{aug}}^{-\frac{1}{2}} \quad \text{with} \quad A_s = \xi_s[\xi_s^\top \xi_s]^{-1}\xi_s^\top \quad \text{and} \quad \xi_s = I_{\text{aug}}^{-\frac{1}{2}} \nabla g_s(\theta^\star). \tag{1.6}$$

In the previous expressions, DM^{EM} and DM^{CM} denote the matrix rate of convergence of the EM and CM (i.e., ECM with no missing data) algorithms respectively. From (1.5)–(1.6), we see that DM^{ECM}, and thus ρ, depends on the order through DM^{CM} where the product of A_s's is calculated in the order that the CM-steps are performed.

Consider the two ECM algorithms, $\text{ECM}_{(1,\ldots,S)}$ with steps ordered: $E \to CM_1 \to \cdots \to CM_S$, and $\text{ECM}_{(S,\ldots,1)}$ with steps ordered: $E \to CM_S \to \cdots \to CM_1$. In general, we will use the notation ECM_α to denote the ECM algorithm with CM-steps ordered as in the ordered set α (e.g., $\alpha = (1,2,4,3)$). The following theorem states that reversing the order of the CM-steps does not change the eigenvalues of DM^{ECM}, a result that is not intuitively clear.

Theorem 1. *For ECM, reversing the CM-steps has no effect on the eigenvalues of* DM^{ECM}.

Proof: From (1.5)–(1.6),

$$[I - DM^{\text{ECM}_{(1,\ldots,S)}}] = I_{\text{obs}}[I_{\text{aug}}^{-1} - I_{\text{aug}}^{-\frac{1}{2}}(A_1 \cdots A_S) I_{\text{aug}}^{-\frac{1}{2}}]. \tag{1.7}$$

Because transposition does not change eigenvalues, (1.7) has the same eigenvalues as

$$\left\{I_{\text{obs}}[I_{\text{aug}}^{-1} - I_{\text{aug}}^{-\frac{1}{2}}(A_1 \ldots A_S) I_{\text{aug}}^{-\frac{1}{2}}]\right\}^\top = \left\{I_{\text{aug}}^{-1} - I_{\text{aug}}^{-\frac{1}{2}}(A_S \ldots A_1) I_{\text{aug}}^{-\frac{1}{2}}\right\} I_{\text{obs}}, \tag{1.8}$$

which in turn has the same eigenvalues as

$$I_{\text{obs}}[I_{\text{aug}}^{-1} - I_{\text{aug}}^{-\frac{1}{2}}(A_S \ldots A_1) I_{\text{aug}}^{-\frac{1}{2}}] = [I - DM^{\text{EM}}][I - DM^{\text{CM}_{(S,\ldots,1)}}] = I - DM^{\text{ECM}_{(S,\ldots,1)}}.$$

□

2. INVESTIGATING STEP ORDERINGS USING LOG-LINEAR MODELS

2.1 REVERSING THE ORDER OF CM-STEPS

From the discussion in Section 1.3, Theorem 1 tells us that reversing CM-steps within the ECM algorithm does not affect ρ and thus by (1.4) it should not affect the number of iterations. We must recall, however, that although ECM tends to be linear near $\theta^\star$, little is known about its behavior away from $\theta^\star$. Moreover, it may take many iterations before $\hat{\rho}_t$ converges to ρ.

To judge the usefulness of Theorem 1, a set of simulations was performed. The ECM algorithm was used to fit a log-linear model to data from a particular partially classified $2 \times 2 \times 2$ contingency table. The model we fitted is a no-three-way-interaction model:

$$\begin{aligned}\log(\theta_{ijk}) = u_0 + (-1)^{i-1}u_1 + (-1)^{j-1}u_2 + (-1)^{k-1}u_3 \\ +(-1)^{i+j}u_{12} + (-1)^{j+k}u_{23} + (-1)^{i+k}u_{13},\end{aligned} \tag{2.1}$$

where θ_{ijk} is the cell probability for cell (i,j,k). Meng and Rubin (1991, 1993) described an ECM algorithm with three CM-steps for this problem. Specifically, because the log-likelihood is linear in the cell counts, $Y = \{y_{ijk}\}$, the E-step simply involves imputing the missing data:

$$y_{ijk}^{(t)} = \tilde{y}_{ijk}^{(a)} + \tilde{y}_i^{(b)} \frac{\theta_{ijk}^{(t)}}{\sum_{jk}\theta_{ijk}^{(t)}} + \tilde{y}_j^{(c)} \frac{\theta_{ijk}^{(t)}}{\sum_{ik}\theta_{ijk}^{(t)}} + \tilde{y}_k^{(d)} \frac{\theta_{ijk}^{(t)}}{\sum_{ij}\theta_{ijk}^{(t)}},$$

Table 1. Simulation Sample Sizes

Sim	n	n_c	n_1	n_2	n_3
1	100	25	25	25	25
2	1000	100	300	300	300
3	100	40	20	20	20

where $\tilde{y}_{ijk}^{(a)}$ are the cell counts for the completely classified observations, and $\tilde{y}_i^{(b)}$, $\tilde{y}_j^{(c)}$, and $\tilde{y}_k^{(d)}$ are the one-way marginal counts. The CM-steps make use of iterated proportional fitting. Given the current estimated cell probabilities $\{\theta_{ijk}^{(t)}\}$, the three CM-steps are

$$\text{CM}_1 : \theta_{ijk}^{(t+\frac{1}{3})} = \theta_{ij(k)}^{(t)} \frac{y_{ij+}}{n}, \quad \text{CM}_2 : \theta_{ijk}^{(t+\frac{2}{3})} = \theta_{i(j)k}^{(t+\frac{1}{3})} \frac{y_{i+k}}{n}, \quad \text{CM}_3 : \theta_{ijk}^{(t+\frac{3}{3})} = \theta_{(i)jk}^{(t+\frac{2}{3})} \frac{y_{+jk}}{n},$$

where n is the total count, $\theta_{ij(k)} = \theta_{ijk} / \sum_k \theta_{ijk}$ is the conditional probability of the third factor given the first two, and $y_{ij+} = \sum_k y_{ijk}$, etc. It is easy to see that CM_1 maximizes $L(\theta|Y)$ subject to $\theta_{ij(1)} = \theta_{ij(1)}^{(t)}$ for each i and j, so that the constraint function $g_1(\theta) = \{\theta_{ij(1)}\}$; likewise $g_2(\theta) = \{\theta_{i(1)k}\}$ and $g_3(\theta) = \{\theta_{(1)jk}\}$.

For each simulation 2,000 data sets of size n were generated, of which n_c were completely classified and n_i were classified only according to margin i ($i = 1, 2, 3$). Three simulations were run and the sample sizes for each appear in Table 1. Each data set was generated from model (2.1). The log-linear parameters were randomly selected for each data set from $u_l \sim N(0, 1)$, and $u_{lm} \sim N(0, .25)$, independently for $l, m = 1, 2, 3$, and u_0 is then chosen so that the cell probabilities sum to one.

For each data set the model in (2.1) was fitted using a starting value of $\theta_{ijk}^{(0)} = \frac{1}{8}$, with each of the six ECM algorithms resulting from the six possible orderings of the CM-steps. Convergence of the algorithm was decided when the standard step-length convergence criterion was met: $||\theta^{(t)} - \theta^{(t-1)}|| \leq 10^{-9}$. The algorithm was allowed to run at most 9,999 iterations. In very sparse data sets (i.e., more than four zero cells), the likelihood surface can be very ill-behaved, in which case ECM may not converge properly. This is of little practical concern because one would not fit model (2.1) with such sparse tables, and has negligible impact on our study since it occurred only 16 times among the 30,000 contingency tables we simulated.

Suppose N_{ij} ($i = 1, 2$; $j = 1, 2, 3$) are the number of steps required for convergence of the six algorithms listed in Table 2, where i and j index the rows and columns respectively. Consider the quantity,

$$R^2 = \frac{\frac{1}{2}\sum_j (N_{1j} - N_{2j})^2}{\sum_{i,j} (N_{ij} - \bar{N}_{\cdot\cdot})^2}, \tag{2.2}$$

which is the proportion of the total variation among the N_{ij} that can be accounted for by

Table 2. ECM Algorithms Used to Fit (2.1) to a 2×2×2 Table

	Cyclic permutations		
Reversals of CM steps	$\text{ECM}_{(123)}$	$\text{ECM}_{(231)}$	7 $\text{ECM}_{(312)}$
	$\text{ECM}_{(321)}$	$\text{ECM}_{(132)}$	$\text{ECM}_{(213)}$

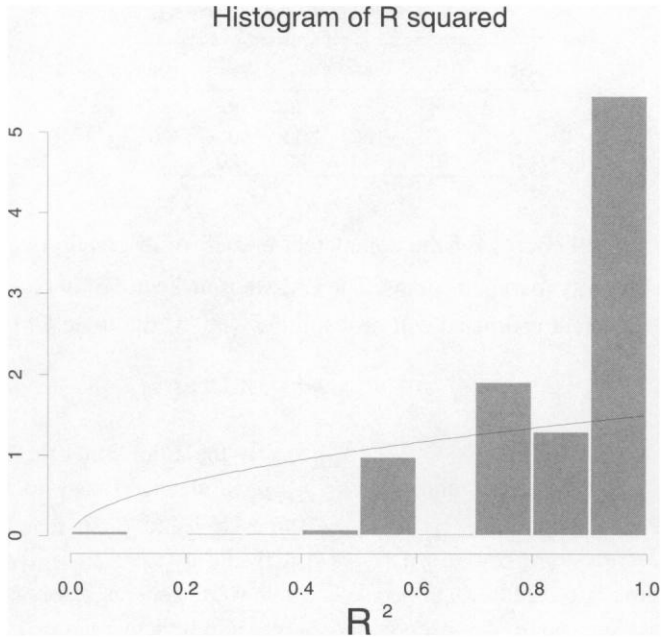

Figure 1. A Histogram of the 4,910 Values of R^2 Observed in the Three Simulations Described in Table 1. Notice that R^2 tends to be larger than expected under the independence assumption (solid line) or according to Theorem 1 ($R^2 \approx 0$).

reversing the CM-steps. If Theorem 1 also implies that reversing the order of CM-steps does not affect the actual number of steps required for convergence, R^2 should be near zero. Figure 1 shows the histogram of the 4,910 simulated values of R^2 for which the denominator of (2.2) was greater than zero. We see that the histogram is actually more skewed to the right than the Beta$(\frac{3}{2}, 1)$ density (solid curve) of R^2 under the assumption that the N_{ij}'s are iid normals; this assumption, suggested by a referee, already represents a serious contradiction to Theorem 1. This indicates that the standard comparison of EM-type algorithms using ρ can be quite misleading, which is not too surprising because ρ only describes the behavior of an EM-type algorithm near convergence.

To explore this further, Figure 2 shows estimated densities for the complement of the possible gain in efficiency due to (a) step reversals (dotted line: $\min(N_{1j}, N_{2j})/\max(N_{1j}, N_{2j})$) and (b) general permutations (solid line: $\min_{i,j}(N_{ij})/\max_{i,j}(N_{ij})$). Notice that although the effect of general permutations is slightly greater than that of step reversals, the latter account for the majority of the gain. This implies that when we consider the effect of permutation of CM-steps on the number of iterations or on CPU time, we must consider all $S!$ possible orderings.

2.2 Factors Affecting Relative Gain

In Figure 2 we see that the relative reduction in the number of steps required by choosing the optimal order is typically less than 20%. But there is a nontrivial portion

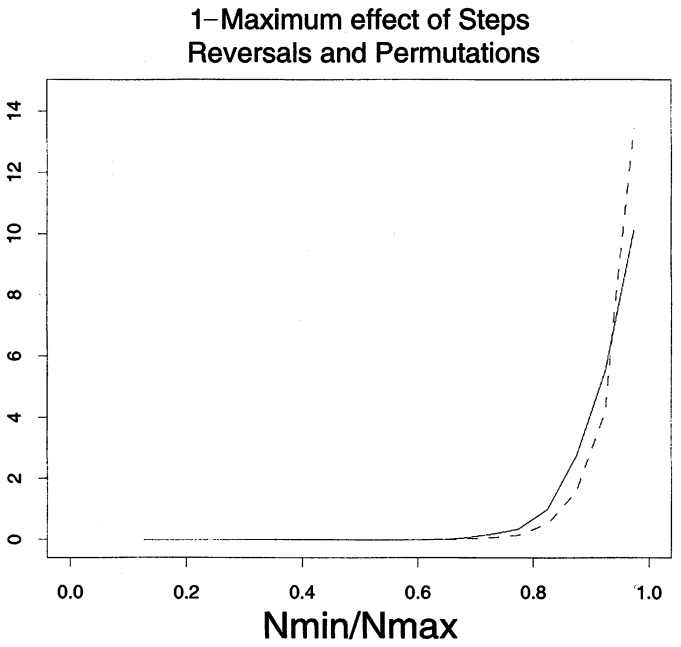

Figure 2. Comparing Step Reversals With General Permutations. The figure shows the density of the effect on iterations required of (a) step reversals: [min(N_{1j},N_{2j})/max(N_{1j},N_{2j})] (dotted line); and (b) general permutation: [min(N_{ij})/max(N_{ij})] (solid line).

of cases where the relative gain is more substantial. In practice, we would like to know what factors will make large improvements likely. One obvious factor is the number of CM-steps—the more steps the more likely that the order will matter; a simulation that confirms this intuition was reported by van Dyk and Meng (1995). In the following we investigate two less obvious factors: the sparseness of the data relative to the number of model parameters and the relative number of incomplete cases.

Using the $2 \times 2 \times 2$ table described in Section 2.1, ten additional simulations were run and are described in Table 3. Simulations 4–8 are designed to look at the effect of sparseness (i.e., $\iota_s \equiv$ (number of parameters)$/n$) and thus the amount of incomplete data was fixed at $\iota_i \equiv 1 - n_c/n = 60\%$ for each of them. The CDF of $N_{\min}/N_{\max}$ for

Table 3. Simulation Sample Sizes and the Resulting Values of the Index of Sparseness, ι_s, and the Index of the Number of Incomplete Cases, ι_i

Sim	n	n_c	n_1	n_2	n_3	$100\times\iota_s$	ι_i
4	40	16	8	8	8	15.00	.60
5	75	30	15	15	15	8.00	.60
6	150	60	30	30	30	4.00	.60
7	500	200	100	100	100	1.20	.60
8	10000	4000	2000	2000	2000	.06	.60
9	100	16	28	28	28	6.00	.84
10	100	31	23	23	23	6.00	.69
11	100	46	18	18	18	6.00	.54
12	100	61	13	13	13	6.00	.39
13	100	76	8	8	8	6.00	.24

each of these simulations appears in the first panel of Figure 3—simulation 4 is on the top and simulation 8 is on the bottom. The effect of sparseness is clear—permutations make more difference in tables with less data. This can be attributed to the fact that in sparse tables the MLE approaches the boundary of the parameter space. As we shall see in Section 3 this pattern persists when random-effects models are fit, in that when the MLE of θ is near the boundary of Θ, the effect of the order of CM-steps is greater. It is reasonable to expect that order is more important when the iterates approach the boundary of the space, and it is in such cases that substantial reduction in the number of iterations is of particular interest as it is known that EM-type algorithms can be painfully slow in such cases (e.g., with a random effects model when the random effects variance is close to zero). This finding suggests that it is more important to study the issue of ordering when the MLE is near the boundary of Θ.

Simulations 9–13 investigate the effect of the number of incomplete cases when the sparseness was fixed at $\iota_s = .06$. The corresponding CDFs of $N_{\min}/N_{\max}$ appear in the second panel of Figure 3. When the data are more incomplete, the CDFs tend to be smoother because ECM requires more steps to converge and thus the integer division in $N_{\min}/N_{\max}$ results in less of a step function. That is, although the CDFs in the bottom panel are shaped rather differently, this seems to be primarily due to the integer

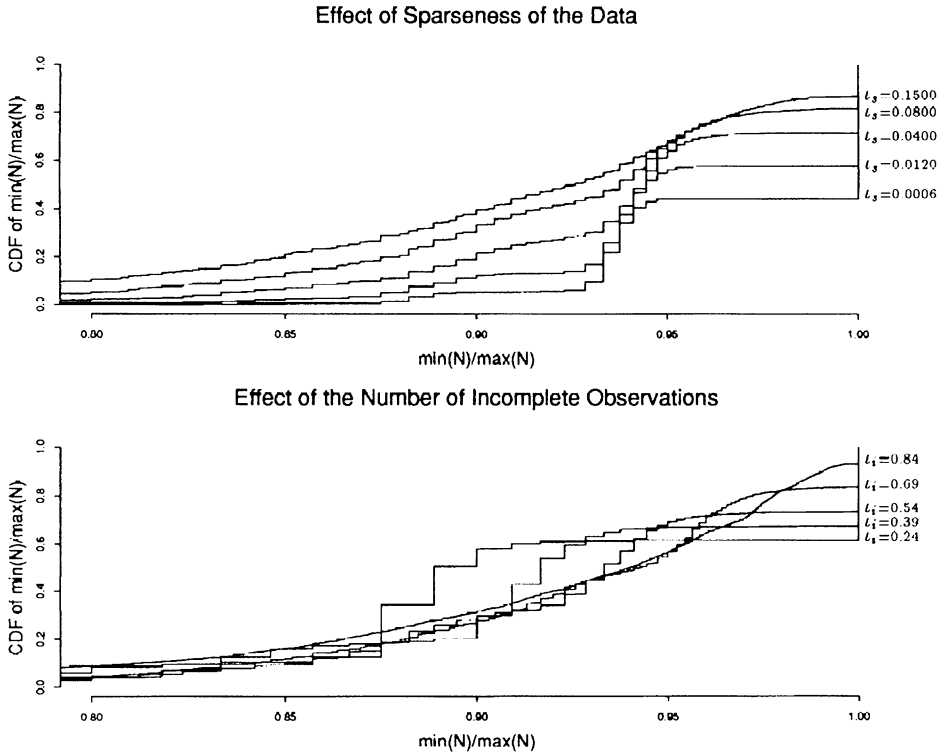

Figure 3. CDFs of $N_{\min}/N_{\max}$ for the Simulations Described in Table 3. The CDFs demonstrate the effect of the sparseness of the data and of the number of incomplete observations.

Table 4. Comparing the Number of Iterations Required for the Cycled ECM Algorithm (N_c) and the Fixed Order ECM Algorithm (N_f) Based on the 6,000 Data Sets Generated in the Simulations Described in Table 1

Sim.	$N_f > N_c$	$N_f = N_c$	$N_f < N_c$
1	.340	.141	.512
2	.447	.095	.459
3	.325	.207	.468

division. The CDFs corresponding to smaller values of ι_i are essentially step-function approximations of the CDF corresponding to $\iota_i = .84$. For example, for 39% of the cases $N_{\min} = N_{\max}$ with $\iota_i = .24$, while for only 7% of the cases did $N_{\min} = N_{\max}$ with $\iota_i = .84$, but the percent of cases for which $.95 \leq N_{\min}/N_{\max}$ is roughly 40% for both $\iota_i = .24$ and $\iota_i = .84$. The *relative* decrease in steps required does not seem to change on average as ι_i increases. It should be noted, however, that when there is more missing information the ECM algorithm is slower so that even with similar relative increase in efficiency, the *absolute* increase in efficiency (i.e., absolute time saved) will increase with ι_i.

2.3 THE CYCLED AND RANDOM ECM ALGORITHMS

If changing the order of CM-steps noticeably affects the rate of convergence of ECM, as is the case when the MLE is near the boundary of the parameter space, we could lose efficiency by making the wrong choice of how to order the CM-steps. Lacking a method for choosing a good order, we might hope to decrease the risk of a badly inefficient algorithm by somehow averaging the orderings. One strategy a practitioner might employ is the "cycled" ECM algorithm described in the following.

Given an ECM algorithm with S CM-step, let $\{\text{ECM}_{\alpha_i}, i = 1, \ldots, S!\}$ be the algorithms resulting from all the possible permutations of the CM-steps. The cycled ECM algorithm runs one iteration from each of these algorithms in an arbitrary order, and then continues to cycle through them until convergence. As will be described in Section 4.1 this can create instability in the step size $||\theta^{(t)} - \theta^{(t-1)}||$, and care must be taken when evaluating convergence of the algorithm. In the simulation, we ran the algorithms until the difference between consecutive log-likelihood values, $L(\theta^{(t+1)}|Y_{\text{obs}}) - L(\theta^{(t)}|Y_{\text{obs}})$, was within a prespecified threshold. Because this requires the evaluation of the actual likelihood, it may not always be feasible in practice, although it is always desirable, since it allows us to check whether the likelihood is increasing at each iteration, a necessary

Table 5. Comparing the Number of Iterations Required for the Random Order ECM Algorithm (N_r) and the Fixed Order ECM Algorithm (N_f) Based on the 6,000 Data Sets Generated in the Simulations Described in Table 1.

Sim.	$N_f > N_r$	$N_f = N_r$	$N_f < N_r$
1	.330	.144	.527
2	.445	.095	.460
3	.323	.218	.460

feature of ECM when it is implemented correctly.

The rationale behind the cycled ECM algorithm is the hope that although it will not be as fast as the fastest ordering of CM-steps, it also should not be as slow as the slowest. It turns out that neither of these is true. The cycled ECM algorithm that incorporated the six ECM algorithms was run on the data generated in the simulations described in Table 1. There were cases in this simulation in which cycled ECM was the fastest, but also cases in which cycled ECM was the slowest. A useful comparison is to compare cycled ECM with a fixed order ECM algorithm. That is, on the outset of running the ECM algorithm, an order for the CM-steps is chosen at random and fixed until convergence (this strategy corresponds to the systematic scan that Liu, Wong, and Kong (1995) studied for implementing the Gibbs sampler). Let the number of iterations required for convergence of the resulting ECM be N_f. This is compared with N_c, the number of iterations required for cycled ECM. For a fair comparison, we define an iteration to be one E-step followed by the three CM-steps for both algorithms (regardless of the order of CM-steps). The results for the three simulations in Table 1 appear in Table 4, and make it clear that on average cycled ECM offers no advantage over haphazard selection of a fixed order of CM-steps, at least in this example.

In what has been described previously, the order in which we cycle through the ECM algorithms is chosen in advance. Instead of doing this, however, we could randomly select an order at each iteration, which corresponds to Liu, Wong, and Kong's (1995) random scan for the Gibbs sampler and was recommended by Amit and Grenander (1991). Given the similarity between ECM and the Gibbs sampler, we hypothesized that random orderings would outperform fixed orderings on average with the ECM algorithm as well. To check our hypothesis we turned to the simulations described in Table 1. We compared the number of iterations required by the algorithm that randomly selects an order at each iteration, N_r, and the algorithm that randomly selects a fixed order on the outset, N_f. The comparison appears in Table 5 and is very similar to Table 4, and thus we have not obtained evidence for the advantage of using these "averaging" strategies instead of simply fixing an arbitrary ordering at the outset. Minimally, our simulation shows that such an advantage, even if it exists, cannot be universal. However, it is possible that more sophisticated strategies, such as selecting different CM-steps with different probabilities, may outperform the fixed-order strategy. We have not investigated these more sophisticated methods, partly because we want to maintain the simplicity of the ECM-type algorithm when we search for faster versions.

3. INVESTIGATING STEP ORDERINGS AND GROUPINGS USING RANDOM-EFFECTS MODELS

3.1 THE ECME IMPLEMENTATIONS

As we mentioned in Section 1.1, in some applications, even though the standard EM algorithm can be easily implemented, the introduction of CM-steps is still useful because the conditional maximizations allow less data augmentation, which can speed up the algorithm. The savings in computational time will depend on the trade-off between using multiple CM-steps, which tends to slow down the algorithm, and using less data-

augmentation, which speeds it up. Furthermore, the efficiency of the algorithm depends on the computations required for each iteration under different implementations. For example, less data augmentation may result in an algorithm that requires fewer iterations, but if each iteration requires iterative computation (e.g., Newton–Raphson) the overall computational time required may be greater. Moreover, even when we have decided how to group the parameters into CM-steps, just as in the contingency-table setting, we still must choose the order in which to run the CM-steps.

All of these issues arise when EM-type algorithms are used to fit random-effects models. In this setting a choice of model-reduction schemes in conjunction with a variety of data-augmentation schemes leads to a multitude of diverse algorithms (e.g., Laird and Ware 1982; Laird, Lange, and Stram 1987; Liu and Rubin 1994; Meng and van Dyk 1997). Here we assume, for illustrative purposes,

$$y_i = X_i^\top \beta + Z_i^\top b_i + e_i, \quad b_i \sim N_q(0, T), \quad e_i \sim N(0, \sigma^2), \quad b_i \perp e_i, \tag{3.1}$$

for $i = 1, \ldots, n$, where X_i $(p \times 1)$ and Z_i $(q \times 1)$ are known covariates (Z_i are such that the model is identifiable); and β are the $(p \times 1)$ fixed effects; $b_i = (b_{i1}, \ldots, b_{iq})^\top$ are the $(q \times 1)$ random effects. (Since the y_i are univariate, (3.1) is essentially a heteroscedastic residuals model; for more general—and more common—mixed-effects models, see Meng and van Dyk (1997).) Although there is no general closed-form solution for the maximum likelihood estimate $\theta^\star \equiv (\beta^\star, \sigma^{2\star}, T^\star)$ of $\theta \equiv (\beta, \sigma^2, T)$ given $Y_{\text{obs}} = (y_1, \ldots, y_n)$, EM-type algorithms provide simple and stable fitting algorithms. For example, Liu and Rubin (1994) presented an ECME algorithm that sets $Q(\theta|\theta^{(t)}) = \mathrm{E}[L(\theta|Y_{\text{aug}}^{(b)})|Y_{\text{obs}}, \theta^{(t)}]$, where $Y_{\text{aug}}^{(b)} = \{(y_i, b_i^\top), i = 1, \ldots, n\}$. The parameter T was updated with the constrained maximizer of $Q(\theta|\theta^{(t)})$, while β and σ^2 were updated with the constrained maximizer of $L(\theta|Y_{\text{obs}})$ (as in (1.3)). Specifically, at the $(t+1)$st iteration,

ECME 1: given $\theta^{(t)} = \left(\beta^{(t)}, \left[\sigma^2\right]^{(t)}, T^{(t)}\right)$
E-step: Calculate for $i = 1, \ldots, n$:

$$\hat{b}_i^{(t+1)} \triangleq \mathrm{E}(b_i|Y_{\text{obs}}, \theta^{(t)}) = \frac{T^{(t)} Z_i (y_i - X_i^\top \beta^{(t)})}{[\sigma^2]^{(t)} + Z_i^\top T^{(t)} Z_i},$$

$$\hat{T}_i^{(t+1)} \triangleq \mathrm{E}(b_i b_i^\top|Y_{\text{obs}}, \theta^{(t)}) = \hat{b}_i^{(t+1)} \left[\hat{b}_i^{(t+1)}\right]^\top + T^{(t)} - \frac{T^{(t)} Z_i Z_i^\top T^{(t)}}{[\sigma^2]^{(t)} + Z_i^\top T^{(t)} Z_i}.$$

CM-step 1: Calculate:

$$T^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \hat{T}_i^{(t+1)};$$

CM-step 2: Calculate:

$$\beta^{(t+1)} = \left\{ \sum_{i=1}^{n} \frac{X_i X_i^\top}{\left[\sigma^2\right]^{(t)} + Z_i^\top T^{(t+1)} Z_i} \right\}^{-1} \left\{ \sum_{i=1}^{n} \frac{X_i y_i}{\left[\sigma^2\right]^{(t)} + Z_i^\top T^{(t+1)} Z_i} \right\}.$$

CM-step 3: Calculate $\left[\sigma^2\right]^{(t+1)}$ as the maximizer of the constrained actual log-likelihood

$$-\frac{1}{2}\sum_{i=1}^{n}\log(\sigma^2 + Z_i^\top T^{(t+1)} Z_i) - \frac{1}{2}\sum_{i=1}^{n}\frac{(y_i - X_i^\top \beta^{(t+1)})^2}{\sigma^2 + Z_i^\top T^{(t+1)} Z_i}.$$

Liu and Rubin (1994) broke up the M-step in this algorithm to increase efficiency, not simplicity. Indeed the global maximizer of $Q(\theta|\theta^{(t)})$ formulated with the data-augmentation used in ECME 1 has a simple closed-form expression (Laird and Ware 1982; Laird et al. 1985). Breaking up the M-step, however, results in an algorithm that requires fewer iterations for convergence (although each iteration is computationally more expensive) because CM-steps 2 and 3 do not use data augmentation.

Meng and van Dyk (1997) suggested an alternative data-augmentation scheme when they developed an EM algorithm in this setting. Let $T = \Delta U \Delta^\top$, where Δ is a lower triangular $(q \times q)$ matrix with all diagonal elements equal to one and U is a diagonal matrix, and set $Y_{\text{aug}}^{(c)} = \{(y_i, c_i^\top), i = 1, \ldots, n\}$, where $c_i = U^{-\frac{1}{2}}\Delta^{-1} b_i$. Model (3.1) can be reexpressed using this data augmentation as

$$y_i = X_i^\top \beta + \sum_{j=1}^{q}\sum_{k=j}^{q} c_{ij} z_{ik} \delta_{kj} u_j + e_i \equiv X_i^\top \beta + \tilde{X}_i^\top \tilde{\beta} + e_i, \tag{3.2}$$

where $c_i = (c_{i1}, \ldots, c_{iq})^\top$, $Z_i = (z_{i1}, \ldots, z_{iq})^\top$, $\{\delta_{jk}, 1 \le j \le k \le q\}$ are the nonzero elements of Δ; $\{u_j, j = 1, \ldots, q\}$ are the diagonal elements of U;

$$\tilde{X}_i = (c_{i1} z_{i1}, \ldots, c_{i1} z_{iq}, c_{i2} z_{i2}, \ldots, c_{i2} z_{iq}, \ldots, c_{iq} z_{iq})^\top, \tag{3.3}$$

and

$$\tilde{\beta} = (\delta_{11} u_1, \ldots, \delta_{q1} u_1, \delta_{22} u_2, \ldots, \delta_{q2} u_2, \ldots, \delta_{qq} u_q)^\top. \tag{3.4}$$

Notice that this data augmentation "transforms" part of the parameter of interest, T, into the missing data, c. This idea plays a key role in Meng and van Dyk's (1997, in press) "working parameter" approach for constructing efficient data-augmentation schemes. They provide both theoretical and empirical evidence to show that using $Y_{\text{aug}}^{(c)} = \{(y_i, c_i^\top), i = 1, \ldots n\}$ instead of $Y_{\text{aug}}^{(b)} = \{(y_i, b_i^\top), i = 1, \ldots n\}$ can lead to substantial savings in computational time when σ^2 is not too small compared with the average of the variances of $Z_i^\top b_i$. Here we combine this idea with ECME—that is, we replace $Y_{\text{aug}}^{(b)}$ with $Y_{\text{aug}}^{(c)}$ when implementing CM-step 1 which leads to

ECME 2: given $\theta^{(t)} = (\beta^{(t)}, \left[\sigma^2\right]^{(t)}, U^{(t)}, \Delta^{(t)})$
E-step: Calculate for $i = 1, \ldots, n$:

$$\hat{c}_i^{(t+1)} \triangleq \mathrm{E}\left[c_i | Y_{\text{obs}}, \theta^{(t)}\right] \equiv \frac{\left(y_i - X_i^\top \beta^{(t)}\right)}{\left[\sigma^2\right]^{(t)} + Z_i^\top \Delta^{(t)} U^{(t)} [\Delta^{(t)}]^\top Z_i} \left[U^{(t)}\right]^{-\frac{1}{2}} \left[\Delta^{(t)}\right]^\top Z_i,$$

$$\hat{B}_i^{(t+1)} \triangleq \mathrm{E}\left[c_i c_i^\top | Y_{\text{obs}}, \theta^{(t)}\right] = \hat{c}_i^{(t+1)} \left[\hat{c}_i^{(t+1)}\right]^\top + I - \frac{\left[U^{(t)}\right]^{-\frac{1}{2}} \Delta^{(t)} Z_i \left[\left[U^{(t)}\right]^{-\frac{1}{2}} \Delta^{(t)} Z_i\right]^\top}{\left[\sigma^2\right]^{(t)} + Z_i^\top \Delta^{(t)} U^{(t)} \left[\Delta^{(t)}\right] Z_i}.$$

CM-step 1: Calculate:

$$\tilde{\beta}^{(t+1)} = \left(\sum_{i=1}^{n} \tilde{B}_i^{(t+1)}\right)^{-1} \sum_{i=1}^{n} \tilde{X}_i^{(t+1)}(y_i - X_i^{\top}\beta^{(t)}),$$

where $\tilde{X}_i^{(t+1)} = \mathrm{E}\left[\tilde{X}_i \middle| Y_{\mathrm{obs}}, \theta^{(t)}\right]$ and $\tilde{B}_i^{(t+1)} = \mathrm{E}\left[\tilde{X}_i\tilde{X}_i^{\top} \middle| Y_{\mathrm{obs}}, \theta^{(t)}\right]$ are simple linear transformations of $\hat{c}_i^{(t+1)}$ and $\hat{B}_i^{(t+1)}$. In particular, they can be calculated using (3.3),

$$\mathrm{E}\left[c_{ij}z_{ik} \middle| Y_{\mathrm{obs}}, \theta^{(t)}\right] = \left[\hat{c}_i^{(t)}\right]_j z_{ik},$$

and

$$\mathrm{E}\left[c_{ij}z_{ik}c_{il}z_{im} \middle| Y_{\mathrm{obs}}, \theta^{(t)}\right] = \left[\hat{B}_i^{(t+1)}\right]_{jl} z_{ik}z_{im},$$

where $\left[\hat{c}_i^{(t)}\right]_j$ is the jth component of the vector $\hat{c}_i^{(t)}$ and $\left[\hat{B}_i^{(t+1)}\right]_{jl}$ is the (j,l)th element of $\hat{B}_i^{(t+1)}$. Using the fact that the diagonal terms of Δ are ones, $\Delta^{(t+1)}$ and $U^{(t+1)}$ can easily be calculated from $\tilde{\beta}^{(t+1)}$ via the relationship given in (3.4). Computationally, the matrix inversion in this CM-step can be avoided by the SWEEP operator (Beaton 1964), as discussed in Little and Rubin (1987, pp. 153–57).

CM-step 2: Calculate $\beta^{(t+1)}$ as in CM-step 2 of ECME 1 by replacing $T^{(t+1)}$ with $\Delta^{(t+1)}U^{(t+1)}\left[\Delta^{(t+1)}\right]^{\top}$;

CM-step 3: Calculate $\left[\sigma^2\right]^{(t+1)}$ as in CM-step 3 of ECME 1 using the same replacement as in CM-step 2.

We note that the SWEEP operator used to update T in ECME 2 produces, as a byproduct, the value of σ^2 that maximizes $Q(\theta|\theta^{(t)})$ conditional on β. If we use this value to update σ^2, we can avoid the iterative CM-step 3 used by ECME 2 and produce an algorithm with a faster ECME iteration. Because we maximize $Q(\theta|\theta^{(t)})$ instead of $L(\theta|Y_{\mathrm{obs}})$ as a function of σ^2, however, more iterations will be required for convergence. The resulting algorithm follows.

ECME 3: given $\theta^{(t)} = (\beta^{(t)}, \left[\sigma^2\right]^{(t)}, U^{(t)}, \Delta^{(t)})$

E-step: This is the same as the E-step of ECME 2;

CM-step 1: Calculate $\tilde{\beta}^{(t+1)}$ and thus $T^{(t+1)} = \Delta^{(t+1)}U^{(t+1)}\left[\Delta^{(t+1)}\right]^{\top}$ as in CM-step 1 of ECME 2 and then calculate

$$\left[\sigma^2\right]^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n}\left[\left(y_i - X_i^{\top}\beta^{(t)} - \left[\tilde{X}_i^{(t+1)}\right]^{\top}\tilde{\beta}^{(t+1)}\right)^2 + \mathrm{tr}\left(\tilde{\beta}^{(t+1)}\left[\tilde{\beta}^{(t+1)}\right]^{\top}\left(\tilde{B}_i^{(t+1)} - \tilde{X}_i^{(t+1)}\left[\tilde{X}_i^{(t+1)}\right]^{\top}\right)\right)\right];$$

CM-step 2: Calculate $\beta^{(t+1)}$ as in CM-step 2 of ECME 2.

ECME 3 groups CM-steps 1 and 3 of ECME 2 into a single CM-step. If we group all the CM-steps into one M-step, maximizing $Q(\theta|\theta^{(t)})$ to update the entire parameter, the resulting algorithm is an EM implementation that was introduced by Meng and van Dyk (1997):

EM: given $\theta^{(t)} = \left(\beta^{(t)}, \left[\sigma^2\right]^{(t)}; \tilde{\beta}^{(t)}\right)$
E-step: This is the same as the E-step of ECME 2.
M-step: Calculate

$$\begin{pmatrix} \beta^{(t+1)} \\ \tilde{\beta}^{(t+1)} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} X_i X_i^\top & \sum_{i=1}^{n} X_i \left[\tilde{X}_i^{t+1}\right]^\top \\ \sum_{i=1}^{n} \tilde{X}_i^{(t+1)} X_i^\top & \sum_{i=1}^{n} \tilde{B}_i^{(t+1)} \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{n} X_i y_i \\ \sum_{i=1}^{n} \tilde{X}_i^{(t+1)} y_i \end{pmatrix}$$

$$\left[\sigma^2\right]^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \left[\left(y_i - X_i^\top \beta^{(t+1)} - \left[\tilde{X}_i^{(t+1)}\right]^\top \tilde{\beta}^{(t+1)} \right)^2 + \mathrm{tr}\left(\tilde{\beta}^{(t+1)} \left[\tilde{\beta}^{(t+1)}\right]^\top \left(\tilde{B}_i^{(t+1)} - \tilde{X}_i^{(t+1)} \left[\tilde{X}_i^{(t+1)}\right]^\top \right) \right) \right].$$

Again, the matrix inversion can be avoided by using the SWEEP operator. Note that this EM implementation is often substantially faster than the standard EM implementation based on $Y_{\text{aug}}^{(b)}$, as demonstrated by Meng and van Dyk (1997).

The four previous algorithms vary in their data-augmentation schemes, how they group the parameters into CM-steps, and the amount of nested iteration that they require. Each of these factors, as well as the order in which the CM-steps are performed, affect the computational time required for convergence. In the following two sections, we will investigate these factors through a series of simulations; note that some of the factors are confounded in our simulation studies due to the nature of these algorithms.

3.2 Step Orderings

In the original presentation of the ECME algorithm, Liu and Rubin (1994) suggested that the CM-steps of ECME, like those of ECM, could be performed in any order. Meng and van Dyk (in press), however, noticed that Liu and Rubin's (1994) proof for the convergence results (e.g., monotone convergence in likelihood) for ECME is valid only when the CM-steps that act on $Q(\theta|\theta^{(t)})$ are performed before those that act on $L(\theta|Y_{\text{obs}})$. Thus, to be sure that the convergence results hold for the ECME algorithms presented in Section 3.1, CM-step 1 must be performed first. Nevertheless, in simulation studies we can implement the algorithm with other orderings to investigate the convergence behavior, including whether the algorithm converges properly when a theoretical guarantee has yet to be established.

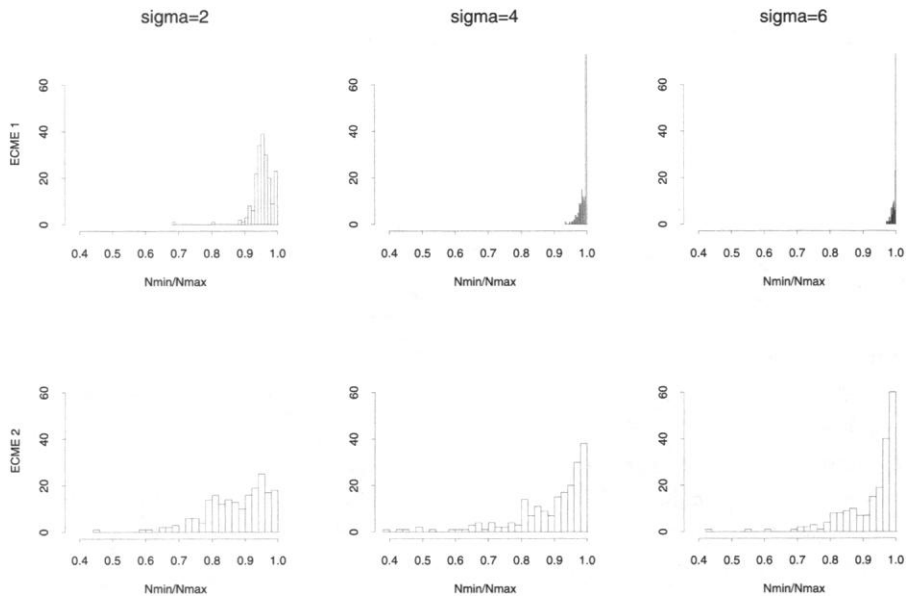

Figure 4. The Effect of Step Permutation on the Number of Iterations Required for Convergence by ECME 1 and ECME 2 for Each of the Three Values of σ. Note that the effect of permutation is more pronounced for both small values of σ and for ECME 2.

We conducted a series of simulations, each with data generated from the model

$$y_i = x_{i1}\beta_1 + x_{i2}\beta_2 + z_{i1}b_{i1} + z_{i2}b_{i2} + e_i, \tag{3.5}$$

where $x_{i1} = 1$, $x_{i2} = i$, z_{ij} were generated independently from $N(0,1)$ at each replication, $\beta_1 = \beta_2 = 1$, $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \sim N_2\left(0, \begin{pmatrix} 4 & 0 \\ 0 & 9 \end{pmatrix}\right)$, and $e_i \sim N(0, \sigma^2)$, with b_i and e_i independent. The simulation was repeated for $\sigma^2 = 4, 16$, and 36. For each of these values, we generated 100 observations from (3.5). The starting values $\beta^{(0)}$ and $[\sigma^2]^{(0)}$ were obtained by fitting (3.5), ignoring the variance components, and $T^{(0)}$ was set to $\begin{pmatrix} 1 & .1 \\ .1 & 1 \end{pmatrix}$. We ran each of the four algorithms presented in Section 3.1 with each of the possible orderings of CM-steps and recorded the number of iterations as well as the computational time required by each algorithm before the log-likelihood convergence criterion $L(\theta^{(t)}|Y_{\text{obs}}) - L(\theta^{(t-1)}|Y_{\text{obs}}) < 10^{-7}$, was reached. The simulation was repeated 200 times, for each of the three σ^2 values.

In all we ran three ECME algorithms on each of 600 data sets. For the first two ECME algorithms there were six possible step orderings and for the third there were two possible orderings. In all cases each algorithm converged to the same point regardless of CM-step orderings. Thus, at least in this setting, we can use all of the possible step orderings despite the lack of theoretical results for some orderings. Figure 4 compares the six possible orderings for the first two ECME algorithms. Let N_{ij} be the number of iterations required for convergence by ECME 1 and ECME 2, where $i = 1, 2$ indexes the algorithm and $j = 1, \ldots, 6$ indexes the order of the CM-steps. Figure 4 displays

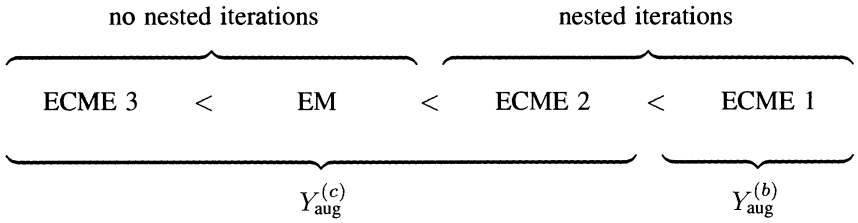

Figure 5. Four ECM-Type Algorithms for Fitting Random-Effects Models. The algorithms differ in their data-augmentation and model-reduction schemes, as well as the need for nested iterations. The inequalities refer to computational time (see Fig. 6).

histograms of $\min_j(N_{ij})/\max_j(N_{ij})$ for ECME 1 and ECME 2 where the data were generated with each of the three values of σ^2. Note that the effect of permutation is much more pronounced with ECME 2 than with ECME 1 and that as with the contingency table example, the effect of permutation dampens when the parameters are further from the boundary of the parameter space (i.e., as σ grows). Also, as in the contingency table example, the empirical distribution (not shown) of R^2 defined in (2.2) was skewed toward 1, so again reversals of CM-steps affected the number of iterations required for convergence (for ECME, Theorem 1 is not applicable).

Of the six CM-step ordering in each of ECME 1 and ECME 2, for two we are assured of monotone convergence in likelihood. For both ECME 1 and ECME 2, these two algorithms were virtually indistinguishable from each other in terms of computational time in this simulation. Moreover, for ECME 1 these two algorithms are generally the fastest to converge; for 73% of the simulated data sets, they were the fastest. Although this pattern did not persist for ECME 2, we still recommend implementing CM-steps that act on $Q(\theta|\theta^{(t)})$ before those that act on $L(\theta|Y_{\text{obs}})$ because these orderings are theoretically superior and empirically performed about the same as the other orderings in our ECME 2 simulations.

3.3 CM-Step Groupings

As we mentioned, more CM-steps will typically result in slower convergence, especially when some of the CM-steps require an iterative solution. On the other hand, more CM-steps allow more flexible augmentation which can result in faster EM-type algorithms. This is the difference between ECME 3 and EM and between ECME 2 and ECME 3 as illustrated in Figure 5. Simulation studies can help us see which of these two forces is stronger. In this section we continue with the simulation described in Section 3.2 and compare the computational time (in seconds) required by each of the three ECME algorithms, which we shall denote τ_1, τ_2, and τ_3, respectively, and that required by EM, τ_{EM}. We must compare computation time rather than simply the number of iterations required for convergence because the iterations of the four algorithms require different computations. To choose the CM-step ordering for the ECME algorithms, we followed the recommendations of the previous section.

The results of this comparison appear in Figures 6 and 7. Figure 6 compares the time required ($\log_{10}$ scale) by the four algorithms (a SUN Sparc 10 was used for the

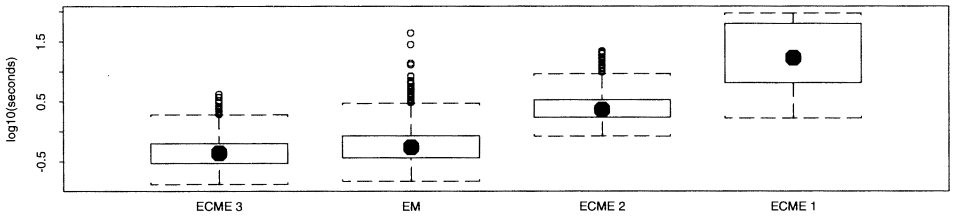

Figure 6. Box Plots for $\log_{10}$ of the Time in Seconds Required by the Four Algorithms Described in Section 3.1 for the Simulations Described in Section 3.2.

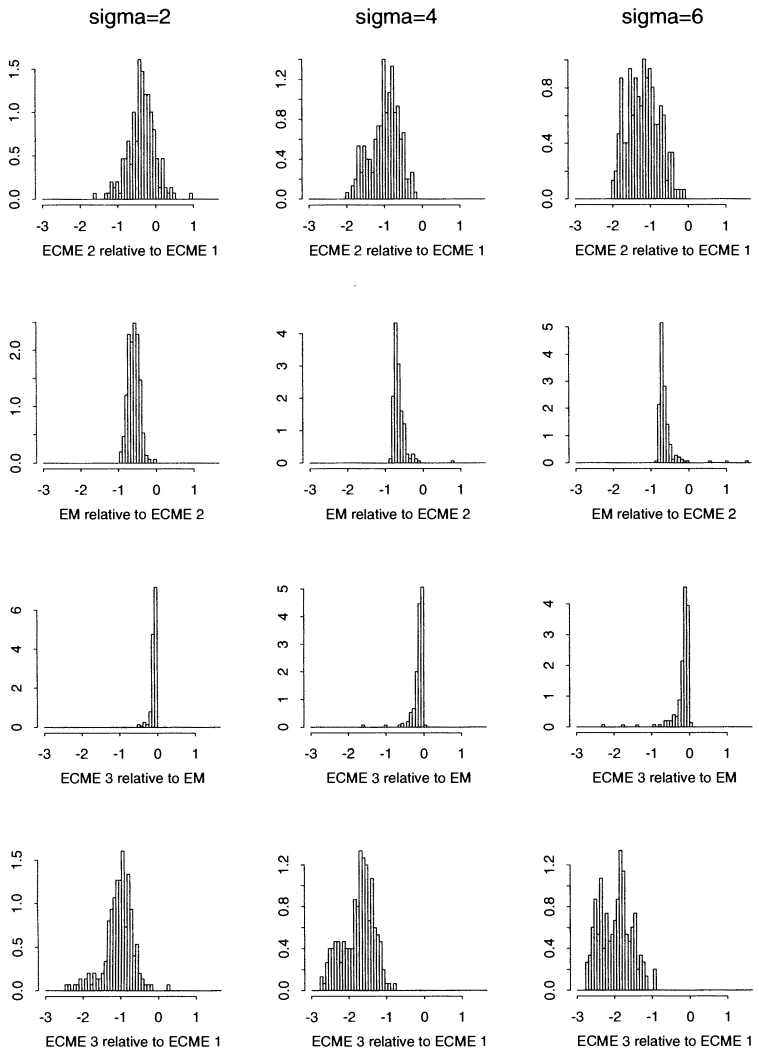

Figure 7. Pairwise Comparisons of the Computation Time Required by the Algorithms Described in Section 3.1. The figure shows histograms of $\log_{10}(\tau_2/\tau_1)$, $\log_{10}(\tau_{EM}/\tau_2)$, $\log_{10}(\tau_3/\tau_{EM})$, and $\log_{10}(\tau_3/\tau_1)$ in each of the rows respectively. The columns correspond to the three values of σ used for generating the data.

computations). It shows that ECME 1, which uses $Y_{\text{aug}}^{(b)}$, is clearly slower than the algorithms that use $Y_{\text{aug}}^{(c)}$. It is also clear that trading reduction in data augmentation for nested iterations as in ECME 1 and ECME 2 is not always a useful strategy, but reducing the model in order to reduce the data augmentation can yield efficient algorithms (i.e., ECME 3 is faster than EM). The relationships between the algorithms are more clearly depicted in Figure 7 which displays pairwise comparisons of the algorithms. The first row compares ECME 2 with ECME 1 (i.e., $\log_{10}(\tau_2/\tau_1)$) and shows that Meng and van Dyk's (1997) data-augmentation scheme improves the algorithms significantly, reducing computation time by a factor of about ten, especially for large σ^2. The second row compares EM and ECME 2 and the third compares ECME 3 with EM. The final row shows the dramatic overall improvement of ECME 3 over ECME 1—computational time is often reduced by a factor of more than 100, especially when σ^2 is large. Moreover, because ECME 3 does not require nested iterations, it is generally easier to implement than either ECME 1 or 2. Thus, for the random-effects model, among these algorithms we recommend ECME 3 with steps ordered as presented in Section 3.1, to ensure monotone convergence in likelihood.

One modification of ECME 3 may be useful when the residual variance is suspected to be very small relative to the variance of the random effects. In this case using $Y_{\text{aug}}^{(b)}$ often results in less data augmentation than using $Y_{\text{aug}}^{(c)}$ and hence a faster algorithm. This can generally be detected in the first few iterations of the algorithm and the data-augmentation scheme switched as described by Meng and van Dyk (1997) in the EM setting.

4. SOME CONCLUDING NOTES

4.1 A REMARK ON A STANDARD CONVERGENCE CRITERION

In all the simulations in Sections 2.1 and 2.2, we used the standard step length criterion: $||\theta^{(t)} - \theta^{(t-1)}||$, which was useful for our purposes both because of its popularity and because it underlies (1.4) which relates the number of steps required for convergence to the spectral radius. In the context of cycled ECM (Section 2.3), however, this criterion can lead to difficulties. If each iteration of cycled ECM is defined as one E-step followed by S CM-steps, then the resulting sequence $\{\theta^{(t)} : t \geq 0\}$ is not a simple linear iteration even at convergence since the mapping that maps $\theta^{(t)}$ to $\theta^{(t+1)}$ changes with t. The difficulty with this is demonstrated in Figure 8, which is a representation of the mapping induced on a subspace of Θ by three ECM algorithms each with a fixed ordering (the smooth curves) and by the cycled ECM algorithm that combines them (the jagged curve). The three ECM algorithms are listed in the first row of Table 2. This is again an example of fitting a log-linear model to a partially classified $2 \times 2 \times 2$ contingency table. The points represent the iteration sequences and show that the cycled ECM algorithm tends to have larger steps. In fact, using the standard convergence criterion, $||\theta^{(t+1)} - \theta^{(t)}|| \leq \epsilon$, this cycled ECM algorithm took 50 times longer to converge than any of the three ECM algorithms. This is in spite of the fact that all the algorithms increased the log-likelihood at about the same rate (cycled ECM is the solid line in the right figure of Figure 8). Because the log-likelihood is increased at each iteration, an alternative convergence criterion is

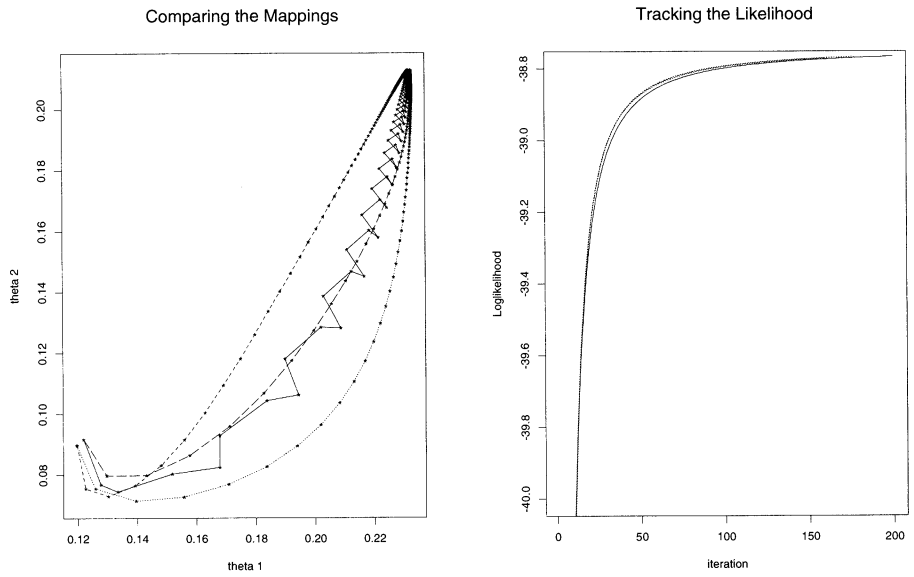

Figure 8. The Cycled ECM Algorithm. The left figure shows the mapping induced on the parameter space (cell probabilities) by three ECM algorithms (dotted lines) and the composite cycle algorithm (solid line). The second figure shows how the four algorithms increase the log-likelihood. The three fixed-order ECM algorithms (upper dashed lines) are indistinguishable and increase the log-likelihood somewhat faster than the cycled algorithm (lower solid line).

$L(\theta^{(t+1)}|Y_{\text{obs}}) - L(\theta^{(t)}|Y_{\text{obs}}) < \epsilon$, which is sensible in the context of likelihood inference (or more generally posterior inference) as discussed in Section 2.3, and thus should always be monitored whenever feasible.

4.2 RECOMMENDATIONS

Our general purpose for investigating the effect of the ordering and grouping of CM-steps is to see if there is a "free and better lunch," not a "better but expensive lunch" in terms of human and computational effort. Given the diversity of ECM-type applications and their diverse model-reduction and data-augmentation schemes, it is impossible to find an "optimal" order-choosing or group-choosing rule that will be universally applicable. Even if such a rule could be found, it has no practical value unless the savings it provides outweigh the cost of implementing it. On the other hand, a practitioner may be interested in knowing about strategies that will lead to relatively efficient algorithms in common implementations of ECM-type algorithms, or at least those that will avoid a very inefficient implementation.

For contingency tables, our investigation shows that it is quite difficult to find a "free and better lunch." Interestingly, however, our investigation reveals some lessons that may have general implications. We not only find that comparing algorithms using the standard theoretical rate of convergence can be quite misleading, but also find that intuitive strategies (e.g., using a random ordering) may yield disappointing results. The recommendation then is simple—do not adopt a strategy that has not been validated by

any realistic empirical evaluation.

For random-effects models, our investigation provides a clear-cut strategy for efficient implementation. Among the algorithms investigated in Section 3, we recommend the ECME 3 algorithm, with CM-steps ordered as in Section 3.1. This algorithm is a blend of efficient model reduction (i.e., few CM-steps, each of which does not require nested iteration) and efficient data augmentation (i.e., small augmented information relative to observed information) and its derivation models how data augmentation and model reduction can be combined to create simple, stable, and fast ECM-type algorithms.

ACKNOWLEDGMENTS

This manuscript was prepared using computer facilities supported in part by NSF Grants DMS 89-05292, DMS 87-05292, and DMS 86-01732 awarded to the Department of Statistics at the University of Chicago; by The University of Chicago Block Fund; and by a MacArthur Fellowship granted to van Dyk by Kalamazoo College. The research was supported in part by NSF Grants DMS 92-04504 and DMS 95-05043. It was also supported in part by the U.S. Census Bureau through a contract with the National Opinion Research Center at the University of Chicago. Meng's research was also supported in part by NSA Grant MDA904-96-1-0007. We thank Y. Amit and J. Fessler for helpful conversations, and a reviewer for comments that led to a much improved presentation.

[Received November 1994. Revised June 1996.]

REFERENCES

Amit, Y., and Grenander, U. (1991), "Comparing Sweep Strategies for Stochastic Relaxation," *Journal of Multivariate Analysis,* 37, 197–222.

Beaton, A. E. (1964), "The Use of Special Matrix Operations in Statistical Calculus," Education Testing Service Research Bulletin, RB-64-51.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood Estimation from Incomplete-Data via the EM Algorithm" (with discussion), *Journal of the Royal Statistical Society,* Ser. B, 39, 1–38.

Fessler, J. A., and Hero, A. O. (1994), "Space-Alternating Generalized Expectation-Maximization Algorithm," *IEEE Transactions on Signal Processing,* 42, 2664–2677.

——— (1995), "Penalized Maximum-Likelihood Image Reconstruction using Space-Alternating Generalized EM Algorithms," *IEEE Transactions on Image Processing,* 4, 1417–1438.

Laird, N., Lange, N., and Stram, D. (1987), "Maximum Likelihood Computations With Repeated Measures: Applications of the EM Algorithm," *Journal of the American Statistical Association,* 82, 97–105.

Laird, N. M., and Ware, J. H. (1982), "Random Effects Models for Longitudinal Data," *Biometrics,* 38, 967–974.

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

Liu, C., and Rubin, D. B. (1994), "The ECME Algorithm: A Simple Extension of ECM With Fast Monotone Convergence," *Biometrika,* 81, 633–648.

Liu, J. S., Wong, W. H., and Kong A. (1994), "Covariance Structures of the Gibbs Sampler With Applications to the Comparisons of Estimators and Augmentation Schemes," *Biometrika,* 81, 27–40.

——— (1995), "Covariance Structure and Convergence Rate of the Gibbs Sampler With Various Scans," *Journal of the Royal Statistical Society,* Ser. B, 57 157–169.

Meng, X. L. (1994), "On the Rate of Convergence of the ECM Algorithm," *The Annals Statistics,* 22, 326–339.

Meng, X. L., and Rubin, D. B. (1991), "IPF for Contingency Tables With Missing Data Via the ECM Algorithm," *Proceedings of the Statistics Computing Section,* Alexandria. VA: American Statistical Association, pp. 244–247.

——— (1992), "Recent Extensions to the EM Algorithm" (with discussion), in *Bayesian Statistics 4,* eds. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, New York: Oxford University Press, pp. 307–320.

——— (1993), "Maximum Likelihood Estimation via the ECM Algorithm: A General Framework," *Biometrika,* 80, 267–278.

——— (1994), "On the Global and Componentwise Rates of Convergence of the EM Algorithm," *Linear Algebra and its Applications* (special issue honoring Ingram Olkin), 199, 413–425.

Meng, X. L., and van Dyk, D. A. (1997), "Fast EM-Type Implementations for Mixed-Effects Models," revised for *Journal of the Royal Statistical Society,* Ser. B.

——— (in press), "The EM Algorithm—An Old Folk Song Sung to a Fast New Tune" (with discussion), *Journal of the Royal Statistical Society,* Ser. B, 59.

Ortega, J. M., and Rheinboldt, W. C. (1970), *Iterative Solutions of Nonlinear Equations in Several Variables,* New York: Academic Press.

van Dyk, D. A., and Meng, X. L. (1995), "Some Findings on the Orderings and Groupings of Conditional Maximizations Within ECM-type Algorithms," technical report No. 397, University of Chicago, Department of Statistics.