Title
Conjugate Gradient Acceleration of the EM Algorithm

Authors
Mortaza Jamshidian
Robert Jennrich

Publication Date
2011-10-24

Conjugate Gradient Acceleration of the EM Algorithm

MORTAZA JAMSHIDIAN and ROBERT I. JENNRICH*

The EM algorithm is a very popular and widely applicable algorithm for the computation of maximum likelihood estimates. Although its implementation is generally simple, the EM algorithm often exhibits slow convergence and is costly in some areas of application. Past attempts to accelerate the EM algorithm have most commonly been based on some form of Aitken acceleration. Here we propose an alternative method based on conjugate gradients. The key, as we show, is that the EM step can be viewed (approximately at least) as a generalized gradient, making it natural to apply generalized conjugate gradient methods in an attempt to accelerate the EM algorithm. The proposed method is relatively simple to implement and can handle problems with a large number of parameters, an important feature of most EM algorithms. To demonstrate the effectiveness of the proposed acceleration method, we consider its application to several problems in each of the following areas: estimation of a covariance matrix from incomplete multivariate normal data, confirmatory factor analysis, and repeated measures analysis. The examples in these areas demonstrate promise for the new acceleration method. In terms of operation counts, for all of the examples considered the accelerated EM algorithm increases the speed of the EM algorithm, in some cases by a factor of 10 or more. In the context of repeated measures analysis, we give a new EM algorithm that, compared to earlier algorithms, can have a considerably smaller cost per iteration. We have not, however, attempted to evaluate the performance of this latter algorithm here.

KEY WORDS: Factor analysis; Incomplete data; Repeated measures.

1. INTRODUCTION

The EM algorithm is a general approach to the computation of maximum likelihood estimates (Dempster, Laird, and Rubin 1977). An attractive feature of EM algorithms is their simplicity in many applications. They are often used as alternatives to the Fisher–scoring and Newton–Raphson algorithms when the latter are too expensive to use or too complicated to implement. A common criticism of EM algorithms, however, is that their convergence can be quite slow (see, for example, Laird, Lange, and Stram 1987; Lindstram and Bates 1988; Horng 1987; Redner and Walker 1984; and several discussants (Nelder, Haberman, Sundberg, and Thompson) of Dempster et al. 1977). For this reason, methods for accelerating the EM algorithm have been proposed.

The most commonly used method for EM acceleration is the multivariate Aitken acceleration

$$\Delta\theta^* = -(\mathbf{J} - \mathbf{I})^{-1}\Delta\theta, \tag{1}$$

where $\Delta\theta$ is the current EM step and $\mathbf{J}$ is the p by p matrix determined by the requirement that it maps each of the p previous EM steps into the succeeding EM step, where p is the number of parameters. Here $\Delta\theta^*$ denotes the accelerated step. Few papers actually have reported on application of this algorithm, and those that did gave mixed reviews (see, for example, Laird et al. 1987; Lindstrom and Bates 1988).

The matrix $\mathbf{J} - \mathbf{I}$ in (1) is an approximation to the Jacobian of the EM step $\Delta\theta$ viewed as a function of the parameter vector θ. Thus (1) is a modified Newton–Raphson algorithm for finding the zeros of $\Delta\theta$. Louis (1982) proposed an algorithm that replaces $\mathbf{J} - \mathbf{I}$ in (1) by the actual Jacobian of the EM step. He called the resulting algorithm an Aitken accelerator. It is, of course, also the Newton–Raphson algorithm for finding the zero of $\Delta\theta$, but Louis did not observe this fact nor have subsequent references to his paper that we have seen.

As Meilijson (1989) observed, however, it is not clear that Louis's algorithm has any advantage over the ordinary Newton–Raphson algorithm for maximum likelihood estimation, and both can be prohibitively expensive on large problems. The primary point of Louis's paper was to give methods for finding the Hessian of the log-likelihood. He used this to find the Jacobian of the EM step. His algorithm was an example given to illustrate the application of his methods; it is not clear that it was intended to be a serious proposal for EM acceleration.

Meilijson (1989) proposed accelerating the EM algorithm by using a quasi-Newton algorithm to find zeros of the EM step, but he did not develop this into a specific proposal and did not report any experience with this suggestion. Meilijson also observed that there are a variety of extrapolation procedures in the numerical analysis literature that may be used for EM acceleration and gave a small example using minimal polynomial extrapolation. The Aitken acceleration (1) may also be viewed as an extrapolator.

On a different tack, Meilijson suggested alternatives to the EM algorithm based on EM ideas or, in his words, "improvement to the EM algorithm on its own terms." His primary alternative was a modified Fisher–scoring algorithm obtained by replacing the Fisher information matrix by an "empirical information matrix" computed from score vectors. The relation to the EM algorithm is that the formulas for the score vectors are derived using EM ideas as proposed

* Mortaza Jamshidian is Assistant Professor, Department of Mathematics, Esfahan University of Technology, Iran. Robert I. Jennrich is Professor, Department of Mathematics, UCLA, Los Angeles, CA 90024. The work of Mortaza Jamshidian was partly supported by National Institute on Drug Abuse Grant DA01070; the work of Robert I. Jennrich was supported by National Science Foundation Grant MCS-8301587. The authors thank the referees for their suggestions and for pointing out the Louis and Meilijson references.

by Louis (1982). Although EM ideas are involved, this does not represent an acceleration method for the EM algorithm, but rather, like the EM algorithm itself, an alternative to the scoring algorithm.

We propose a general acceleration method that, in keeping with the usual EM algorithm, is fairly simple and deals conveniently with problems having many parameters. The latter is an area where the EM algorithm is particularly important and where it often is the only algorithm used. An important problem of this type is estimating a covariance matrix from incomplete normal data. As the title suggests, our method will use conjugate gradients. More specifically, we will treat an EM step as a generalized gradient and use a generalized conjugate gradient algorithm as an EM accelerator. Though not identified as such, this was done by Golub and Nash (1982) in their alternative to the Yates (1933) EM algorithm for fitting unbalanced analysis of variance models.

We will attempt to show by examples that the general conjugate gradient approach proposed can be used effectively in a variety of application areas. The areas considered include estimating the covariance matrix from incomplete data, confirmatory factor analysis, and repeated measures analysis. These are some applications where the EM algorithm is important and acceleration is often needed. In some of the examples we have found reductions of a factor of 10 or more in the operation count.

Section 2 defines the generalized conjugate gradient algorithm that we used, and Section 3 explains why it is natural to view EM steps as generalized gradients and, specifically, how they are to be used in the algorithm of Section 2. Section 4 presents examples.

2. A GENERALIZED CONJUGATE GRADIENT ALGORITHM

We give here an algorithm for finding the maximum of a function $f(\theta)$, where θ ranges over a subset of Euclidian p space. Let $\mathbf{g}(\theta)$ denote the gradient of $f(\theta)$ and consider the generalized norm

$$\|\theta\| = (\theta^T \mathbf{W} \theta)^{1/2} \tag{2}$$

on Euclidian p space defined by a positive definite matrix $\mathbf{W}$. Let $\tilde{\mathbf{g}}(\theta)$ be the gradient of $f(\theta)$ with respect to this norm. This is a fancy way of saying that

$$\tilde{\mathbf{g}}(\theta) = \mathbf{W}^{-1}\mathbf{g}(\theta). \tag{3}$$

The vector $\tilde{\mathbf{g}}(\theta)$ is called the generalized gradient of $f(\theta)$ defined by $\mathbf{W}$.

The use of appropriate generalized gradients can significantly improve the performance of algorithms that use gradients. Indeed, this choice can be the most important choice made in the formulation of an algorithm. We will discuss our choice of generalized gradients in the next section. The generalized conjugate gradient algorithm that we will use proceeds as follows:

Given θ_0, let $\mathbf{d}_0 = \tilde{\mathbf{g}}_0$ and sequentially compute:

$$\alpha_k, \text{ the value of } \alpha \text{ that maximizes } f(\theta_k + \alpha \mathbf{d}_k), \tag{4}$$

$$\theta_{k+1} = \theta_k + \alpha_k \mathbf{d}_k \tag{5}$$

$$\beta_k = \tilde{\mathbf{g}}_{k+1}^T(\mathbf{g}_{k+1} - \mathbf{g}_k)/\mathbf{d}_k^T(\mathbf{g}_{k+1} - \mathbf{g}_k), \tag{6}$$

and

$$\mathbf{d}_{k+1} = \tilde{\mathbf{g}}_{k+1} - \beta_k \mathbf{d}_k, \tag{7}$$

where $\mathbf{g}_k = \mathbf{g}(\theta_k)$ and $\tilde{\mathbf{g}}_k = \tilde{\mathbf{g}}(\theta_k)$. This is called a generalized conjugate gradient algorithm, because it uses generalized gradients to define the search directions $\mathbf{d}_k$ and because, for negative definite quadratic functions f, the $\mathbf{d}_k$ are orthogonal in the metric defined by the negative of the Hessian of f.

The choice of the update formula (6) is somewhat arbitrary. It differs from the more popular Fletcher–Reeves (1964) and Polak–Riebere (1969) updates. It has the advantage that for quadratic $f(\theta)$, $\mathbf{d}_k$ and $\mathbf{d}_{k+1}$ are conjugate even if the line searches used in (4) are not exact. What is also important is that the algorithm requires only gradients and generalized gradients and does not require explicit computation of generalized inner products.

The algorithm defined by (4) through (7) will converge to a maximum of a quadratic function in at most p steps. For algorithms with this property, it is common practice to restart every p steps or more often. We do the former, which means setting $\mathbf{d}_k = \tilde{\mathbf{g}}_k$ whenever k is an integer multiple of p.

3. ACCELERATING THE EM ALGORITHM

The EM algorithm of Dempster et al. (1977) is designed to find a parameter vector $\hat{\theta}$ that maximizes a likelihood function $L(\theta)$, $\theta \in \Theta$. For a specific application, a function $Q(\theta', \theta)$ is identified. It may be viewed as a local approximation to $\log L(\theta')$ in a neighborhood of θ. (For an explicit definition of $Q(\theta', \theta)$, see (A.1) in the Appendix.) Let $\tilde{\theta}$ be the value of θ' that maximizes $Q(\theta', \theta)$; then the step $\tilde{\theta} - \theta$ is called an EM step. We show in the Appendix that if $\hat{\theta}$ is an interior point of Θ, then

$$\tilde{\theta} - \theta = -(\ddot{Q}(\hat{\theta}, \hat{\theta}))^{-1}\mathbf{g}(\theta) + o(\theta - \hat{\theta}), \tag{8}$$

where $\mathbf{g}(\theta)$ is the gradient of $\log L(\theta)$ at θ and $\ddot{Q}(\hat{\theta}, \hat{\theta})$ is the Hessian of $Q(\theta', \theta)$ viewed as a function θ' and evaluated at $(\theta', \theta) = (\hat{\theta}, \hat{\theta})$. Typically $\ddot{Q}(\hat{\theta}, \hat{\theta})$ is negative definite. Thus using (8), when θ is near $\hat{\theta}$, the EM step $\tilde{\theta} - \theta$ is, to a good approximation, a generalized gradient of $\log L(\theta)$.

We propose attempting to accelerate the EM algorithm by using EM steps, $\tilde{\theta} - \theta$, as the generalized gradients $\tilde{\mathbf{g}}(\theta)$ in the algorithm of the previous section with $f(\theta) = \log L(\theta)$. We call the resulting algorithm the accelerated EM (AEM) algorithm. Note that this is not strictly speaking an application of the algorithm of the previous section, but rather an approximation to the algorithm of that section that uses

$$\mathbf{W} = -\ddot{Q}(\hat{\theta}, \hat{\theta}). \tag{9}$$

The advantage of the approximation, of course, is that it does not require $\hat{\theta}$ and in fact does not require $\mathbf{W}$. Each step begins with an EM step. First, its direction is modified, and then its length is optimized. The next section will look at the effectiveness of the proposed acceleration.

Although the algorithm proposed is fairly simple, it is more complex than the EM algorithm itself. In addition to the EM steps, one must compute gradients $\mathbf{g}(\theta)$ of $\log L(\theta)$. Frequently, however, these require only a simple modification of the EM code. The biggest complication is the line search required in (4). Our algorithm for this is given in the

Appendix. Our experience is that a simple line search is sufficient.

4. NUMERICAL EXAMPLES AND COMPARISONS

We illustrate the acceleration method of the previous section by looking at several examples in each of the three areas identified in the introduction. For each area we give formulas for $f(\theta) = \log L(\theta)$, for its gradient $\mathbf{g}(\theta)$, and for the EM step $\tilde{\theta} - \theta$. In most cases the EM step can be expressed simply in terms of $\mathbf{g}(\theta)$, and in such cases it is this form that will be given. Throughout we have attempted to implement our formulas with reasonably efficient code, but there is of course always a trade-off between efficiency and simplicity. Our rule of thumb for the examples considered has been that we will accept an estimated 10% loss of efficiency in return for simplicity of implementation. We feel that this is a reasonable compromise for our purpose and that it has had little effect on the comparisons we have made.

We start each AEM algorithm with a few EM steps. More specifically, given a starting value we continue taking EM steps as long as the difference between two successive values of 2 log $L(\theta)$ is greater than 1; that is, as long as the χ^2 statistic for testing the equality of two successive iterates is more than 1. As soon as this condition is violated, we start applying the AEM algorithm. For our examples and our starting values, in most cases the AEM algorithm started after about five steps of the EM algorithm. To compare the speed of the AEM algorithm to the EM algorithm, we use the number of floating point operations (FLOPs) required from the point where the AEM algorithm is started to the point where we obtain six significant digits of accuracy in the value of 2 log L. Iterations are counted in the same manner. The FLOPs are counted by the PC version of the matrix language MATLAB. This is the language that we have used to code our algorithms. For the examples considered, we have noted that six significant digits of accuracy in the value of 2 log L roughly corresponds to two or three significant digits of accuracy in the values of the parameters.

4.1 Estimation of a Covariance Matrix From Incomplete Data

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a sample from a p-dimensional normal distribution with covariance matrix $\mathbf{\Sigma}$. We assume that some of the components of each $\mathbf{x}_i$ may not be observed. Let $\mathbf{y}_i$ be the subvector of $\mathbf{x}_i$ containing the observed components. We wish to estimate $\mathbf{\Sigma}$ from the density of the observed $\mathbf{y}_i$. In the interest of simplicity, we assume that the distribution sampled has mean 0. Let $\mathbf{\Sigma}_i$ denote the covariance matrix for $\mathbf{y}_i$. Then the log-likelihood at $\mathbf{\Sigma}$ is

$$f = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}(\log|\mathbf{\Sigma}_i| + \mathbf{y}_i^T\mathbf{\Sigma}_i^{-1}\mathbf{y}_i) \quad (10)$$

and its gradient is

$$G = \frac{1}{2}\sum_{i=1}^{n}[\![\mathbf{\Sigma}_i^{-1}(\mathbf{y}_i\mathbf{y}_i^T - \mathbf{\Sigma}_i)\mathbf{\Sigma}_i^{-1}]\!], \quad (11)$$

where the expression $[\![A]\!]$ is A padded with 0s to make a p by p matrix so that A is the same submatrix of $[\![A]\!]$ as $\mathbf{\Sigma}_i$ is of $\mathbf{\Sigma}$. The usual formula for the EM update is

$$\tilde{\mathbf{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} E(\mathbf{x}_i\mathbf{x}_i^T|\mathbf{y}_i, \mathbf{\Sigma}). \quad (12)$$

As shown in the Appendix, we may express the EM step as

$$\tilde{\mathbf{\Sigma}} - \mathbf{\Sigma} = \frac{2}{n}\mathbf{\Sigma}G\mathbf{\Sigma}. \quad (13)$$

This shows, as predicted by (8), that the EM step is an approximate generalized gradient when $\mathbf{\Sigma}$ is near its maximum likelihood estimate $\hat{\mathbf{\Sigma}}$. It also suggests why the EM algorithm converges very slowly when $\hat{\mathbf{\Sigma}}$ is nearly singular.

In applying the EM and AEM algorithms, we have taken θ as the upper triangular part of $\mathbf{\Sigma}$ and have defined $\mathbf{g}(\theta)$ as the upper triangular part of (11). Also, $\tilde{\mathbf{g}}$ is set to the upper triangular part of (13).

For numerical comparisons, we report results of three examples in Table 1. Example 1 used incomplete real data from a rheumatoid arthritis study (Spiegel et al. 1986), with 92 cases and 8 variables. The variables are four repeated measures of the 50-foot walk time in seconds for each patient and the number of assistance devices, ranging from 0 to 16, that he or she uses on each walk. Each variable was adjusted by subtracting its mean. Example 2 used artificial data generated from a multivariate normal distribution with mean 0 and covariance matrix $\mathbf{I}$, the identity, with $n = 30$ and $p = 10$. In this example two 5 by 5 matrices are missing from the data matrix, one from the top right and one from the bottom left. The EM algorithm is reasonably efficient for these two examples; nevertheless, the AEM algorithm accelerated the EM algorithm by a factor of 3.8 on Example 1 and a factor of 3.7 on Example 2.

The data for Example 3 are given in Table 2. For this example $\hat{\mathbf{\Sigma}}$ is nearly singular and the EM algorithm converges very slowly. The value of 2 log L was correct up to four significant digits after 174 iterations of the EM algorithm, as compared to after four iterations of the AEM algorithm. In terms of the number of FLOPs, this is a factor of more than 13 in favor of the AEM algorithm. As shown in Table 1, after 2,000 iterations the EM algorithm did not have the value of 2 log L correct up to six significant digits. Based on the number of FLOPs, the AEM algorithm exhibited a speed at least 92 times faster than the EM algorithm to obtain six significant digits of accuracy. This example demonstrates the potential for spectacular gains in applications where the EM algorithm is known to converge slowly (see Horng 1987).

A heuristic explanation for the superior performance of the AEM algorithm might proceed as follows: The eigen-

Table 1. Comparison of the EM Algorithm With the AEM Algorithm on Three Covariance Matrix Estimation Examples

Example	EM FLOPs[a]	AEM FLOPs	EM/AEM[b]
1	59.2 (59)[c]	15.6 (7)	3.8
2	35 (74)	9.5 (7)	3.7
3	>22 (>2,000)	.24 (8)	>92

[a] The number of FLOPs shown is the actual number of FLOPs divided by 10^6. The count starts as defined in the text.
[b] The ratio of the number of FLOP's for the EM algorithm to that for the AEM algorithm.
[c] The number of iterations.

Table 2. Data Used in Covariance Matrix Estimation Example 3

y_1	y_2	y_3
−11/8	−3/8	*[a]
−3/8	5/8	*
5/8	13/8	*
−3/8	−3/8	*
*	−11/8	−1/4
*	−3/8	3/4
*	5/8	7/4
*	−3/8	−1/4
−3/8	*	3/4
5/8	*	−1/4
13/8	*	−5/4
−3/8	*	−5/4

[a] Value was not observed.

values of the Hessian H of $\log L$ in the metric of $\hat{\Sigma} \otimes \hat{\Sigma}$ determine the asymptotic rate of convergence of both the EM and AEM algorithms. If $\hat{\Sigma}$ is nearly singular, but has only a few small eigenvalues, then we expect H to have a few large, but mostly moderate, eigenvalues. The AEM algorithm (conjugate gradient) performs well under these conditions, but the EM algorithm (steepest descent) does not (see, for example, Luenberger 1984, p. 246).

4.2 Confirmatory Factor Analysis

We consider the factor analysis model

$$\mathbf{y} = \Lambda\mathbf{f} + \mathbf{e}, \tag{14}$$

where $\mathbf{y}$ is a vector of observed values, Λ is a p by $k < p$ matrix of factor loadings, and $\mathbf{f}$ and $\mathbf{e}$ are independent normally distributed random vectors with mean 0 and covariance matrices Φ and Ψ. By assumption Ψ is diagonal. Following common practice and for simplicity, we have assumed that the mean of $\mathbf{y}$ is 0. The covariance matrix of $\mathbf{y}$ is

$$\Sigma = \Lambda\Phi\Lambda^T + \Psi. \tag{15}$$

We allow a priori restrictions that fix arbitrary elements of Λ and Ψ at specified values and, following Rubin and Thayer (1982), we allow Φ to be set equal to the identity or be totally free.

Given n independent observations $\mathbf{y}_i$, let $S = \sum_{i=1}^{n} \times \mathbf{y}_i\mathbf{y}_i^T/n$. Given S, the log-likelihood of (Λ, Φ, Ψ) is

$$f = -\frac{n}{2}(p \log 2\pi + \log|\Sigma| + \operatorname{tr} S\Sigma^{-1}). \tag{16}$$

The problem is to compute maximum likelihood estimates $(\hat{\Lambda}, \hat{\Phi}, \hat{\Psi})$.

Rubin and Thayer (1982) described the EM algorithm for obtaining $(\hat{\Lambda}, \hat{\Phi}, \hat{\Psi})$ using $\binom{y}{f}$ as the complete data. At each step the EM algorithm updates (Λ, Φ, Ψ) as follows: Solve the system of linear equations

$$[\![\tilde{\Lambda}B - S\Sigma^{-1}\Lambda\Phi]\!]_\Lambda = 0 \tag{17}$$

for $\tilde{\Lambda}$. Here $[\![A]\!]_\Lambda$ is simply A with 0s inserted in the places corresponding to the fixed parameters in Λ and

$$B = \Phi + \Phi\Lambda^T\Sigma^{-1}(S - \Sigma)\Sigma^{-1}\Lambda\Phi. \tag{18}$$

Using $\tilde{\Lambda}$,

$$\tilde{\Psi} - \Psi = [\![S - \Psi - 2S\Sigma^{-1}\Lambda\Phi\tilde{\Lambda}^T + \tilde{\Lambda}B\tilde{\Lambda}^T]\!]_\Psi, \tag{19}$$

where $[\![\cdot]\!]_\Psi$ is defined in a manner similar to $[\![\cdot]\!]_\Lambda$. Finally, if Φ is free, then

$$\tilde{\Phi} = B; \tag{20}$$

otherwise, $\Phi = I$ and is not a parameter.

The derivatives of f are given by

$$\frac{\partial f}{\partial \Lambda} = n\Sigma^{-1}(S - \Sigma)\Sigma^{-1}\Lambda\Phi, \tag{21}$$

$$\frac{\partial f}{\partial \Phi} = \frac{n}{2}\Lambda^T\Sigma^{-1}(S - \Sigma)\Sigma^{-1}\Lambda, \tag{22}$$

and

$$\frac{\partial f}{\partial \Psi} = \frac{n}{2}\operatorname{diag}[\Sigma^{-1}(S - \Sigma)\Sigma^{-1}]. \tag{23}$$

The gradient $\mathbf{g}$ of f is formed by selecting the elements from these matrices corresponding to free parameters, and $\tilde{\mathbf{g}}$ is formed by selecting from $\tilde{\Lambda} - \Lambda$, $\tilde{\Phi} - \Phi$, and $\tilde{\Psi} - \Psi$.

Table 3 shows the result of using the EM and AEM algorithms on three examples. The data for Example 1 are based on the nine psychological tests exploration sample first analyzed by Holzinger and Swineford (1939). We use the model used by Hägglund (1982). This model has three factors and 33 free parameters. For starting values we use Hägglund's FABIN3 estimates, reported in his Tables 1 through 3. On this example the AEM algorithm was 2.7 times as fast as the EM algorithm as shown in Table 3.

Examples 2 and 3 are artificial data problems with $p = 15$, $k = 4$, and $n = 300$. They both use the same model. The matrix $\Lambda = \binom{I}{\Lambda_1}$, where I is a 4 by 4 identity matrix and Λ_1 is a free 11 by 4 matrix. All elements of Φ and all the diagonal elements of Ψ are assumed to be free. This structure was used to produce two moderate-sized problems with a relatively large number of identified parameters: 69 in each case.

To generate the data for Example 2, random numbers were selected in the interval [0, 3] for the components of Λ_1, and the matrices Φ and Ψ were taken as identity matrices. These were used to compute Σ, which in turn was used to generate 300 random normal vectors $\mathbf{y}_i$. The latter were the data for Example 2. As shown in Table 3, the AEM algorithm was 10 times faster than the EM algorithm for this example.

The data for Example 3 were generated like those for Example 2, except that the elements of Φ were set to $\phi_{ij} = .7^{|i-j|}$. We chose this because it seems that the performance of the

Table 3. Comparison of the EM Algorithm With the AEM Algorithm on Three Confirmatory Factor Analysis Examples

Example	EM FLOPs[a]	AEM FLOPs	EM/AEM[b]
1	1.6 (21)[c]	.6 (4)	2.7
2	100 (333)	9.9 (17)	10
3	143 (473)	12 (21)	12

[a] The number of FLOPs shown is the actual number of FLOPs divided by 10^6. The count starts as defined in the text.

[b] The ratio of the number of FLOPs for the EM algorithm to that for the AEM algorithm.

[c] The number of iterations.

EM algorithm is sensitive to the spread of the eigenvalues of $\hat{\Phi}$. The more spread the eigenvalues of $\hat{\Phi}$, the slower the EM algorithm converges. As shown in Table 3, the AEM algorithm converged reasonably rapidly and beat the EM algorithm by a factor of 12. The starting values for Examples 2 and 3 were the values of Λ, Φ, and Ψ used to generate the data $\mathbf{y}_i$.

4.3 The Repeated Measures Model

Here we consider a general linear mixed model for repeated measures. More specifically, let $\mathbf{y}_i$ denote a t_i by 1 vector of t_i measurements observed on the ith of a set of experimental units. We use the two-stage mixed model considered by Laird and Ware (1982):

$$\mathbf{y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{b}_i + \mathbf{e}_i, \tag{24}$$

where $\mathbf{X}_i$ and $\mathbf{Z}_i$ are known t_i by p and t_i by q design matrices, β is a vector of fixed effects to be estimated, and $\mathbf{b}_i$ and $\mathbf{e}_i$ are independent random vectors distributed as $\mathcal{N}(0, D)$ and $\mathcal{N}(0, \rho I_i)$. Here $\mathbf{D}$ is a positive definite q by q matrix, $\rho > 0$, and $\mathbf{I}_i$ is a t_i by t_i identity matrix. We seek estimates for β, $\mathbf{D}$, and ρ.

For this model,

$$E(\mathbf{y}_i) = X_i\beta$$

and

$$\operatorname{var}(\mathbf{y}_i) = \Sigma_i = Z_i D Z_i^T + \rho I_i. \tag{25}$$

The log-likelihood is

$$f = -\frac{1}{2}\left[T\log 2\pi + \sum_{i=1}^{n}(\log|\Sigma_i| + \mathbf{r}_i^T\Sigma_i^{-1}\mathbf{r}_i)\right],$$

where $T = \sum_{i=1}^{n} t_i$ and $\mathbf{r}_i = \mathbf{y}_i - X_i\beta$. The derivatives of f with respect to β, D, and ρ are

$$\frac{\partial f}{\partial \beta} = \sum_{i=1}^{n} X_i^T\Sigma_i^{-1}\mathbf{r}_i, \tag{26}$$

$$\frac{\partial f}{\partial D} = \frac{1}{2}\sum_{i=1}^{n} Z_i^T\Sigma_i^{-1}(\mathbf{r}_i\mathbf{r}_i^T - \Sigma_i)\Sigma_i^{-1}Z_i, \tag{27}$$

and

$$\frac{\partial f}{\partial \rho} = \frac{1}{2}\sum_{i=1}^{n}\operatorname{tr}[\Sigma_i^{-1}(\mathbf{r}_i\mathbf{r}_i^T - \Sigma_i)\Sigma_i^{-1}]. \tag{28}$$

Laird and Ware (1982) gave an iterative algorithm for computing maximum likelihood estimates of β, D, and ρ. Given current values of ρ and D, they computed

$$\tilde{\beta} = \left(\sum_{i=1}^{n} X_i^T\Sigma_i^{-1}X_i\right)^{-1}\sum_{i=1}^{n} X_i^T\Sigma_i^{-1}\mathbf{y}_i \tag{29}$$

and then used

$$\Delta\rho = \frac{2\rho^2}{T}\frac{\widetilde{\partial f}}{\partial \rho} \tag{30}$$

and

$$\Delta D = \frac{2}{n}D\frac{\widetilde{\partial f}}{\partial D}D \tag{31}$$

to update ρ and D. Here $\widetilde{\partial f}/\partial\rho$ and $\widetilde{\partial f}/\partial D$ are (27) and (28) evaluated at $(\tilde{\beta}, D, \rho)$. The formulas (30) and (31) are a reformulation of the Laird and Ware (LW) algorithm that is derived in the Appendix. This algorithm was called a hybrid EM algorithm by Jennrich and Schluchter (1986). This name is appropriate, because at each step the value of β is computed by generalized least squares and then, assuming that β is fixed at its new value, the EM algorithm is applied to update values of D and ρ.

The EM algorithm for this problem using $\binom{\mathbf{y}_i}{\mathbf{b}_i}$ as the complete data is also derived in the Appendix. The steps at a point (β, D, ρ) are

$$\Delta\beta = \rho\left(\sum_{i=1}^{n} X_i^T X_i\right)^{-1}\frac{\partial f}{\partial \beta}, \tag{32}$$

$$\Delta D = \frac{2}{n}D\frac{\partial f}{\partial D}D, \tag{33}$$

and

$$\Delta\rho = \frac{1}{T}\left(2\rho^2\frac{\partial f}{\partial \rho} - \rho\,\Delta\beta^T\frac{\partial f}{\partial \beta}\right). \tag{34}$$

Surprisingly, we could not find this EM algorithm in the literature. It beat the LW algorithm in two out of three of our experiments. This is mainly because the LW algorithm requires computation of $(\sum_{i=1}^{n} X_i^T\Sigma_i^{-1}X_i)^{-1}$ every iteration, as opposed to the EM algorithm, which requires instead the computation of $(\sum_{i=1}^{n} X_i^T X_i)^{-1}$ only once.

It is difficult to conclude anything about the relative performance of the EM and LW algorithms from our limited examples. The first example, which we will describe shortly, involves complete data. Generalized and ordinary least squares are identical for this example. As a consequence, both algorithms make identical steps; but as noted the LW algorithm uses formulas that are more expensive to evaluate.

It is the EM algorithm defined by (32) through (34) that we will attempt to accelerate, but we include the Laird and Ware algorithm for the purpose of comparison. Equations (26) through (28) define the elements of $\mathbf{g}$ and (32) through (34) define the elements of $\tilde{\mathbf{g}}$ for the AEM algorithm.

For our examples we use the data given by Pothoff and Roy (1964) and Model 6 given by Jennrich and Schluchter (1986). Starting values for ρ and D are computed as described in Section 4.4 of Laird et al. (1987). The starting values of β for the EM and AEM algorithms were obtained by ordinary least squares. The LW algorithm does not require a starting value for β.

In Example 1 we used the Pothoff and Roy (1964) data with no missing values. As shown in Table 4, for this example the AEM algorithm is 13 times faster than the LW algorithm and 6.5 times faster than the EM algorithm.

Example 2 uses the data given by Little and Rubin (1987, p. 159), which is the Pothoff and Roy (1964) data with some observations missing. For this example AEM is 2.5 times faster than EM and 1.6 times faster than LW; these are rather modest factors. This is also an example where the LW algorithm did better than the EM algorithm by a factor of about 1.5.

Table 4. Comparison of the EM, LW, and AEM Algorithms on Three Repeated Measures Examples

Example	EM FLOPs[a]	LW FLOPs	AEM FLOPs	LW/AEM[b]	EM/AEM
1	20.0 (51)[c]	10 (51)	1.54 (4)	13	6.5
2	2.6 (8)	3.8 (23)	1.60 (5)	1.6	2.5
3	89.0 (387)	43.6 (387)	1.8 (7)	49.4	24.2

[a] The number of FLOPs shown is the actual number of FLOPs divided by 10^5. The count starts as defined in the text.
[b] The ratio of the number of FLOPs for the LW algorithm to that for the AEM algorithm.
[c] The number of iterations.

Finally, Example 3 uses the Pothoff and Roy (1964) data with one observation deleted at random per subject. Again from Table 4, the EM and LW algorithms required the same number of iterations; however, in terms of the number of FLOPs, EM is twice as fast as LW. We have no explanation why both algorithms used the same number of iterations. They did not follow the same path. Also in this example, AEM converges considerably faster than EM and LW; specifically, it beats EM by a factor of 24 and LW by a factor of 49.

5. DISCUSSION

The EM algorithm often works well, which explains its popularity. What we have done here is to attempt to extend the range of its applicability without sacrificing too much of the simplicity it usually enjoys. The EM algorithm is particularly important for problems involving a large number of parameters; for such problems, it is often the only algorithm used. The conjugate gradient method, which is simple and does not require storage or inversion of large matrices, is a particularly natural method to use as an acceleration device for the EM algorithm.

There are, however, many problems that do not involve large numbers of parameters; for those, one might consider replacing the conjugate gradient algorithm, considered here, by a quasi-Newton algorithm. Because for quadratic functions the generalized conjugate gradient algorithm is the Davidon–Fletcher–Powell algorithm using ordinary gradients, but started with W^{-1} as the initial inverse Hessian approximation, this would be a natural choice of quasi-Newton algorithm. In our present context we do not have W explicitly, but assuming that one has both ordinary and generalized gradients, the explicit use of W can be avoided by using (3).

As noted previously, Meilijson has suggested using a quasi-Newton algorithm for finding the 0s of the EM step; but some care is required. Because the Jacobian of the EM step is in general not symmetric, familiar symmetric quasi-Newton updates, such as the one given by Meilijson (1989), will not work. One solution is to use general nonsymmetric updates (see, for example, Broyden 1972), but this ignores the fact that the EM step is an approximate generalized gradient and hence its Jacobian has special structure. A better alternative, we suspect, is one that uses updates that are self-adjoint in the metric of W. There are many options here. As far as we know, none of these have been tried.

Finally, we have derived a new EM algorithm for the repeated measures problem. Its cost per iteration can be considerably less than that of the LW algorithm, but our experience with the two algorithms is too limited to recommend one over the other. The EM algorithm proposed can be made simpler still by dropping the term involving $\Delta\beta$ from (34). This is motivated by the fact that near a solution, the dropped term will be very small. It would be natural to consider conjugate gradient acceleration of both the LW algorithm and that just identified. We have had some success in accelerating the LW algorithm, but in keeping with our title, we have restricted our report here to the acceleration of the EM algorithm.

It should be clear that the acceleration methods proposed here need not be restricted to the EM algorithm. One might try these methods on any algorithm where steps are, approximately at least, generalized gradients. They might for example be used to accelerate the scoring algorithm, because its steps are approximately generalized gradients in the metric of the Fisher information matrix evaluated at $\hat{\theta}$. We have looked at a number of such alternative applications in the context of factor analysis (Jamshidian and Jennrich 1988), with some success.

APPENDIX: ACCELERATION OF THE EM ALGORITHM

Here we give several derivations promised in the text and our line search algorithm.

A.1 Derivation of Equation (8)

Equation (8) follows from Theorem 1 on page 219. Given a family of densities $f_1(\mathbf{y}|\theta)$, $\theta \in \Theta$, the EM algorithm is a device for finding a maximum likelihood estimate $\hat{\theta}$ for θ based on $\mathbf{y}$. It begins by introducing a function $\mathbf{h}$ and a second family of densities $f_2(\mathbf{x}|\theta)$, $\theta \in \Theta$ related to the first by the requirement that if a random vector $\mathbf{x}$ has density $f_2(\mathbf{x}|\theta)$, then the random vector $\mathbf{y} = \mathbf{h}(\mathbf{x})$ has density $f_1(\mathbf{y}|\theta)$. The algorithm is determined by the choice of the second family $f_2(\mathbf{x}|\theta)$. The vector $\mathbf{x}$ is usually referred to as the complete data, and $\mathbf{y}$ is referred to as the observed, or incomplete, data.

The algorithm begins by defining the function

$$Q(\theta', \theta) = E[\log f_2(\mathbf{x}|\theta')|\mathbf{y}, \theta]. \tag{A.1}$$

Given θ, a step of the EM algorithm creates a new vector $\tilde{\theta}$ such that $\theta' = \tilde{\theta}$ maximizes $Q(\theta', \theta)$. By replacing θ with $\tilde{\theta}$, the algorithm produces a sequence of values of θ that hopefully converges to a maximum likelihood estimate. Dempster et al. (1977) and Wu (1983) have given sufficient conditions for this convergence. Usually one attempts to choose the complete data, so it is simple to maximize $Q(\theta', \theta)$ with respect to θ'. In what follows we assume that there is a function A that generates $\tilde{\theta}$ from θ so that $\tilde{\theta} = A(\theta)$, and that $\hat{\theta}$ is a fixed point of A. Let $\dot{Q}(\theta', \theta)$ and $\ddot{Q}(\theta', \theta)$ denote the gradient and Hessian of $Q(\theta', \theta)$ with respect to its first argument, and let $\mathbf{s}(\theta) = \partial(\log f_1(\mathbf{y}|\theta))/\partial\theta$ be the Fisher score vector for the observed data.

Lemma 1. Let $\mathcal{O}$ be a convex open subset of Θ containing $\hat{\theta}$. If (a) $Q(\theta', \theta)$ is twice continuously differentiable with respect to θ' for all θ' and θ in $\mathcal{O}$, (b) A has a Jacobian at $\hat{\theta}$, and (c) $\dot{Q}(\theta, \theta) = \mathbf{s}(\theta)$ for all $\theta \in \mathcal{O}$, then for θ near $\hat{\theta}$,

$$\mathbf{s}(\theta) + \ddot{Q}(\hat{\theta}, \hat{\theta})(A(\theta) - \theta) = o(\theta - \hat{\theta}). \tag{A.2}$$

Proof. Let $\theta \in \mathcal{O}$ and $\tilde{\theta} = A(\theta)$. Also, let $\dot{Q}_i(\theta', \theta)$ be the ith component of $\dot{Q}(\theta', \theta)$ and let $\ddot{Q}_i(\theta', \theta)$ be the ith row of $\ddot{Q}(\theta', \theta)$.

By the mean value theorem, there is a $\bar{\theta}$ between θ and $\tilde{\theta}$ such that

$$\dot{Q}_i(\tilde{\theta}, \theta) = \dot{Q}_i(\theta, \theta) + \ddot{Q}_i(\bar{\theta}, \theta)(\tilde{\theta} - \theta). \quad (A.3)$$

Because $\theta' = \tilde{\theta}$ maximizes $Q(\theta', \theta)$ with respect to θ',

$$\dot{Q}_i(\tilde{\theta}, \theta) = 0. \quad (A.4)$$

Let $s_i(\theta)$ be the ith element of $\mathbf{s}(\theta)$. Using (A.4) and (c), it follows from (A.3) that

$$\begin{aligned} 0 &= s_i(\theta) + \ddot{Q}_i(\bar{\theta}, \theta)(\tilde{\theta} - \theta) \\ &= s_i(\theta) + \ddot{Q}_i(\hat{\theta}, \hat{\theta})(\tilde{\theta} - \theta) \\ &\quad + (\ddot{Q}_i(\bar{\theta}, \theta) - \ddot{Q}_i(\hat{\theta}, \hat{\theta}))(\tilde{\theta} - \theta). \end{aligned} \quad (A.5)$$

Let $\hat{J}$ be the Jacobian of A at $\hat{\theta}$; then

$$\tilde{\theta} - \theta = (\hat{J} - I)(\theta - \hat{\theta}) + o(\theta - \hat{\theta}). \quad (A.6)$$

By the continuity of $\ddot{Q}_i$,

$$\ddot{Q}_i(\bar{\theta}, \theta) - Q_i(\hat{\theta}, \hat{\theta}) \to 0, \quad (A.7)$$

as $\theta \to \hat{\theta}$. Using (A.6) and (A.7), (A.5) takes the form

$$0 = s_i(\theta) + \ddot{Q}_i(\hat{\theta}, \hat{\theta})(\tilde{\theta} - \theta) + o_i(\theta - \hat{\theta}). \quad (A.8)$$

The lemma follows from the fact that (A.8) holds for each $i = 1, \ldots, p$.

Theorem 1. If, in addition to the assumptions of Lemma 1, $\ddot{Q}(\hat{\theta}, \hat{\theta})$ is negative definite, then for θ near $\hat{\theta}$,

$$A(\theta) - \theta = -(\ddot{Q}(\hat{\theta}, \hat{\theta}))^{-1}\mathbf{s}(\theta) + o(\theta - \hat{\theta}). \quad (A.9)$$

Proof. Apply Lemma 1.

Equation (8) is obtained by noting that $\mathbf{g}(\theta) = \mathbf{s}(\theta)$. Assumption (c) is a technical condition that often holds. Fisher (1925) showed that

$$\mathbf{s}(\theta) = E\left(\frac{\partial}{\partial\theta} \log f_2(\mathbf{x}|\theta)|\mathbf{y}, \theta\right). \quad (A.10)$$

Assuming that the derivative can be removed from under the conditional expectation leads to (c).

A.2 Derivation of Equation (13)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and $\mathbf{y}_1, \ldots, \mathbf{y}_n$ be defined as in Section 4.1, and let $f_2(\mathbf{x}|\boldsymbol{\Sigma}')$ denote the joint density of the complete data $\mathbf{x}_1, \ldots, \mathbf{x}_n$. It is easy to verify that

$$\frac{\partial}{\partial\boldsymbol{\Sigma}'} f_2(\mathbf{x}|\boldsymbol{\Sigma}') = \frac{1}{2}\sum_{i=1}^{n} \boldsymbol{\Sigma}'^{-1}(\mathbf{x}_i\mathbf{x}_i^T - \boldsymbol{\Sigma}')\boldsymbol{\Sigma}'^{-1}. \quad (A.11)$$

Taking conditional expectation given the observed data $\mathbf{y}_1, \ldots, \mathbf{y}_n$ and $\boldsymbol{\Sigma}$, evaluating the result at $\boldsymbol{\Sigma}' = \boldsymbol{\Sigma}$ and using (A.10) gives

$$G = G(\boldsymbol{\Sigma}) = \frac{1}{2}\sum_{i=1}^{n} \boldsymbol{\Sigma}^{-1}(E(\mathbf{x}_i\mathbf{x}_i^T|\mathbf{y}_i, \boldsymbol{\Sigma}) - \boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1}. \quad (A.12)$$

But using (12),

$$G = \frac{n}{2}\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1}. \quad (A.13)$$

Solving for $\tilde{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}$ in (A.13) gives equation (13).

A.3 Derivations of Equations (30) and (31)

Laird and Ware (1982) updated ρ by the equation

$$\tilde{\rho} = \frac{1}{T}\sum_{i=1}^{n} E(\|\mathbf{y}_i - X_i\tilde{\boldsymbol{\beta}} - Z_i\mathbf{b}_i\|^2|\mathbf{y}_i, \theta). \quad (A.14)$$

Let $f_2(\mathbf{x}|\theta')$ denote the joint density of the complete data $\binom{\mathbf{y}_i}{\mathbf{b}_i}$; $i = 1, \ldots, n$, with $\theta' = (\tilde{\boldsymbol{\beta}}, \rho', D')$, where $\tilde{\boldsymbol{\beta}}$ is defined by (29). It is easy to verify that

$$\frac{\partial}{\partial\rho'} \log f_2(\mathbf{x}|\theta') = -\frac{T}{2\rho'} + \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{y}_i - X_i\tilde{\boldsymbol{\beta}} - Z_i\mathbf{b}_i\|^2/\rho'^2. \quad (A.15)$$

Taking conditional expectation with respect to the observed data $\mathbf{y}_1, \ldots, \mathbf{y}_n$ and using Fisher's result (A.10) gives

$$\frac{\partial\tilde{f}}{\partial\rho} = -T/2\rho + T\tilde{\rho}/2\rho^2, \quad (A.16)$$

where $\partial\tilde{f}/\partial\rho$ is the derivative (28) evaluated at $(\tilde{\boldsymbol{\beta}}, \rho, D)$. Solving (A.16) gives

$$\tilde{\rho} - \rho = \frac{2\rho^2}{T}\frac{\partial\tilde{f}}{\partial\rho}, \quad (A.17)$$

which is (30). Equation (31) follows from a similar, but simpler, argument.

A.4 Derivation of Equations (32), (33), and (34)

Let $f_2(\mathbf{x}|\theta')$ be as in the previous section, but let $\theta' = (\boldsymbol{\beta}', \rho', D')$. It is easy to verify that

$$\frac{\partial}{\partial\boldsymbol{\beta}'} \log f_2(\mathbf{x}|\theta') = \sum_{i=1}^{n} X_i^T(\mathbf{y}_i - X_i\boldsymbol{\beta}' - Z_i\mathbf{b}_i)/\rho'. \quad (A.18)$$

Let θ denote the current value of θ and let $\tilde{\theta}$ be its value after an EM step. Taking conditional expectation given $\mathbf{y}_1, \ldots, \mathbf{y}_n$ and θ, evaluating the result at $\theta' = \tilde{\theta}$ and at $\theta' = \theta$, and using Fisher's result (A.10) gives

$$0 = \sum_{i=1}^{n} X_i^T(\mathbf{y}_i - X_i\tilde{\boldsymbol{\beta}} - Z_i\mathbf{b}_i^*) \quad (A.19)$$

and

$$\rho\frac{\partial f}{\partial\boldsymbol{\beta}} = \sum_{i=1}^{n} X_i^T(\mathbf{y}_i - X_i\boldsymbol{\beta} - Z_i\mathbf{b}_i^*), \quad (A.20)$$

where $\mathbf{b}_i^* = E(\mathbf{b}_i|\mathbf{y}_i, \theta)$. Solving (A.19) and (A.20) gives

$$\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta} = \rho\left(\sum_{i=1}^{n} X_i^TX_i\right)^{-1}\frac{\partial f}{\partial\boldsymbol{\beta}}, \quad (A.21)$$

which is Equation (32).

It is easy to verify that $\partial \log f_2(\mathbf{x}|\theta')/\partial\rho'$ is (A.15) with $\tilde{\boldsymbol{\beta}}$ replaced by $\boldsymbol{\beta}'$. Taking conditional expectations and evaluating at $\theta' = \tilde{\theta}$ and at $\theta' = \theta$ as before gives

$$0 = -T\tilde{\rho} + \sum_{i=1}^{n} (\|\mathbf{y}_i - X_i\tilde{\boldsymbol{\beta}} - Z_i\mathbf{b}_i\|^2)^* \quad (A.22)$$

and

$$2\rho^2\frac{\partial f}{\partial\rho} = -T\rho + \sum_{i=1}^{n} (\|\mathbf{y}_i - X_i\boldsymbol{\beta} - Z_i\mathbf{b}_i\|^2)^*, \quad (A.23)$$

where $(\cdot)^*$ denotes the conditional expectation of $(\cdot)$. Using (A.19), the summation on the right side of (A.23) may be written as

$$\sum_{i=1}^{n} (\|\mathbf{y}_i - X_i\tilde{\boldsymbol{\beta}} - Z_i\mathbf{b}_i\|^2)^* + \sum_{i=1}^{n} \|X_i(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2. \quad (A.24)$$

Using this and solving (A.22) and (A.23) gives

$$2\rho^2\frac{\partial f}{\partial\rho} = T(\tilde{\rho} - \rho) + \sum_{i=1}^{n} \|X_i(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2. \quad (A.25)$$

Using (A.21) gives

$$\tilde{\rho} - \rho = \frac{1}{T}\left(2\rho^2\frac{\partial f}{\partial\rho} - \rho(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})^T\frac{\partial f}{\partial\boldsymbol{\beta}}\right), \quad (A.26)$$

which is equation (34). Equation (33) follows by a similar, but simpler, argument.

A.5 A Line Search Algorithm

Let θ be a given point in a p-dimensional Euclidian space, and let $\mathbf{d}$ be a given p-dimensional vector. Here we give the algorithm

used in the examples of Section 4 to locate the maximum of

$$F(\alpha) = f(\theta + \alpha \mathbf{d}), \qquad \alpha \geq 0. \tag{A.27}$$

The slope of $F(\alpha)$ at α is given by

$$\dot{F}(\alpha) = \mathbf{d}^T \mathbf{g}_\alpha, \tag{A.28}$$

where $\mathbf{g}_\alpha$ is the gradient of f evaluated at $\theta + \alpha \mathbf{d}$. We assume that $F(\alpha)$ and $\mathbf{g}_\alpha$ are defined at $\alpha = 0$ and that $\dot{F}(0) \geq 0$. To obtain the maximizing α, we look for the solution of the equation $\dot{F}(\alpha) = 0$ using the secant method (see, for example, Johnson and Riess 1982, p. 166). This method, along with some added details, is as follows:

Step 0: Set $\alpha_0 = 0$, $\alpha_1 = 2$, and $n = 0$, compute $\dot{F}(0)$, and set $\dot{F}(\alpha_0) = \dot{F}(0)$.

Step 1: If $\dot{F}(\alpha_1)$ is defined, then
- compute $\dot{F}(\alpha_1)$
- set $n = n + 1$.

Otherwise:
- if $n = 10$, quit
- set $\alpha_1 = \alpha_1/2$
- set $n = n + 1$
- go to Step 1.

Step 2: If $n = 10$, quit. If $n \neq 1$ and $|\dot{F}(\alpha_1)| < .1\dot{F}(0)$, then accept α_1 as the minimizing value and go to the next iteration.
Otherwise, if $\text{sign}[(\alpha_1 - \alpha_0)](\dot{F}(\alpha_0) - \dot{F}(\alpha_1))/(|\dot{F}(\alpha_0)| + |\dot{F}(\alpha_1)|) < 10^{-5}$, then quit;
Otherwise:
- set $\alpha^* = (\alpha_1 \dot{F}(\alpha_0) - \alpha_0 \dot{F}(\alpha_1))/(\dot{F}(\alpha_0) - \dot{F}(\alpha_1))$
- set $\alpha_0 = \alpha_1$
- set $\dot{F}(\alpha_0) = \dot{F}(\alpha_1)$
- set $\alpha_1 = \alpha^*$
- go to Step 1.

As indicated in Steps 1 and 2, we allow at most 10 and require at least two function-gradient evaluations per line search. In our examples the mode for n was 2. We feel that limiting n to 2 will have little, if any, effect on the convergence of the AEM algorithm. We did not encounter $n = 10$ or a "true" response for the second "if" in Step 2. Had we encountered either of these, we would have quit the search and restarted the AEM algorithm. As indicated in Step 2, α is accepted as a good approximation to the maximizing α, if $|\dot{F}(\alpha)| < .1\dot{F}(0)$. The value .1 is, of course, our choice and is one which in our experience seems to give a good enough approximation to the maximizing value. The choice $\alpha_1 = 2$ in Step 0 is natural, because it corresponds to a double-length EM step.

[*Received January 1990. Revised May 1992.*]

REFERENCES

Broyden, C. G. (1972), "Quasi-Newton Methods," in *Numerical Methods for Unconstrained Optimization,* ed. W. Murray, New York: Academic Press, pp. 87–106.

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

Fisher, R. A. (1925), "Theory of Statistical Estimation," *Proceedings of the Cambridge Philosophical Society,* 22, 700–725.

Fletcher, R., and Reeves, C. M. (1964), "Function Minimization by Conjugate Gradients," *Computer Journal,* 7, 149–154.

Golub, G. H., and Nash, S. G. (1982), "Nonorthogonal Analysis of Variance Using a Generalized Conjugate-Gradient Algorithm," *Journal of the American Statistical Association,* 77, 109–116.

Hägglund, G. (1982), "Factor Analysis by Instrumental Variables Methods," *Psychometrika,* 47, 209–222.

Holzinger, K. J., and Swineford, F. (1939), *A Study in Factor Analysis: the Stability of a Bi-Factor Solution,* Supplementary Educational Monographs, No. 48, Chicago: University of Chicago Press.

Horng, S. C. (1987), "Examples of Sublinear Convergence of the EM Algorithm," in *Proceedings of the Statistical Computing Section, American Statistical Association,* pp. 266–271.

Jamshidian, M., and Jennrich, R. I. (1988), "Conjugate Gradient Methods in Confirmatory Factor Analysis," *UCLA Statistics Series,* No. 8.

Jennrich, R. I., and Schluchter, M. D. (1986), "Unbalanced Repeated-Measures Models with Structured Covariance Matrices," *Biometrics,* 42, 805–820.

Johnson, L. W., and Riess, R. D. (1982), *Numerical Analysis,* Reading, MA: Addison-Wesley.

Laird, N. M., Lange, N., and Stram, D. O. (1987), "Maximum Likelihood Computations with Repeated Measures: Application 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, 963–974.

Lindstrom, M. J., and Bates, D. M., (1988), "Newton–Raphson and EM Algorithms for Linear Mixed-Effects Models for Repeated-Measures Data," *Journal of the American Statistical Association,* 83, 1014–1022.

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

Louis, T. A. (1982), "Finding the Observed Information Matrix When Using the EM Algorithm," *Journal of the Royal Statistical Society,* Ser. B, 44, 226–233.

Luenberger, D. G. (1984), *Linear and Nonlinear Programming,* Reading, MA: Addison-Wesley.

Meilijson, I. (1989), "A Fast Improvement to the EM Algorithm on Its Own Terms," *Journal of the Royal Statistical Society,* Ser. B, 51, 127–138.

Polak, E., and Ribiere, G. (1969), "Note sur la Convergence de Methods de Directions Conjugres," *Revue Francaise Informat. Recherche Operationnelle,* 16, 35–43.

Pothoff, R. F., and Roy, S. N. (1964), "A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems," *Biometrika,* 51, 313–326.

Redner, R. A., and Walker, H. F. (1984), "Mixture Densities, Maximum Likelihood, and the EM Algorithm," *SIAM Review,* 26, 195–239.

Rubin, D. B., and Thayer, D. T. (1982), "EM Algorithms for ML Factor Analysis," *Psychometrika,* 47, 69–76.

Spiegel, J. S., Spiegel, T. M., Ward, N. B., Paulus, H. E., Leake, B., and Kane, R. L. (1986), "Rehabilitation for Rheumatoid Arthritis Patients, A Controlled Trial," *Arthritis and Rheumatism,* 29, 628–637.

Wu, C. F. Jeff (1983), "On the Convergence Properties of the EM Algorithm," *The Annals of Statistics,* 11, 95–103.

Yates, F. (1933), "The Analysis of Replicated Experiments When the Field Results Are Incomplete," *Empire Journal of Experimental Agriculture,* 1, 129–142.