Statistics and Computing **13**: 137–152, 2003
© 2003 Kluwer Academic Publishers. *Manufactured in The Netherlands.*

The one-step-late PXEM algorithm

DAVID A. VAN DYK and RUOXI TANG

Department of Statistics, Harvard University

Received December 2001 and accepted September 2002

The EM algorithm is a popular method for computing maximum likelihood estimates or posterior modes in models that can be formulated in terms of missing data or latent structure. Although easy implementation and stable convergence help to explain the popularity of the algorithm, its convergence is sometimes notoriously slow. In recent years, however, various adaptations have significantly improved the speed of EM while maintaining its stability and simplicity. One especially successful method for maximum likelihood is known as the parameter expanded EM or PXEM algorithm. Unfortunately, PXEM does not generally have a closed form M-step when computing posterior modes, even when the corresponding EM algorithm is in closed form. In this paper we confront this problem by adapting the one-step-late EM algorithm to PXEM to establish a fast closed form algorithm that improves on the one-step-late EM algorithm by insuring *monotone* convergence. We use this algorithm to fit a probit regression model and a variety of dynamic linear models, showing computational savings of as much as 99.9%, with the biggest savings occurring when the EM algorithm is the slowest to converge.

Keywords: dynamic linear model, EM algorithm, MAP estimates, one-step-late methods, PXEM algorithm, posterior modes, probit regression, rate of convergence, working parameters

1. Introduction

Computing maximum likelihood (ML) or maximum a posteriori (MAP) estimates in highly structured models involving latent variables, missing data, random effects, etc., can be a challenging computational task. The EM algorithm (Dempster, Laird and Rubin 1997), however, has proved to be a powerful tool in such complex settings. The primary advantages of the EM algorithm over Newton-Raphson-type algorithms are numerical and computational stability (see Dempster, Laird and Rubin 1997, Meng and van Dyk 1997). In particular, the EM algorithm is often easy to implement and is known to increase the loglikelihood or log posterior function at each iteration. Unfortunately, in some cases the EM algorithm can be very slow to converge. Thus, in recent years a number of extensions and adaptations of the algorithm have been proposed to improve its rate of convergence while maintaining its simplicity and stability (e.g., Liu and Rubin 1994, van Dyk 2000b, c). An especially promising set of algorithms for ML involves the method of working parameters (Meng and van Dyk 1997) and the PXEM algorithm (Liu, Rubin and Wu 1998). These EM-type algorithms and their stochastic counterparts have been used to significantly improve computational speed in a wide range of models (Liu and Rubin 1994, 1995, Gelfand, Sahu and Carlin 1995, Meng and van Dyk 1997, 1998, 1999, Higdon 1998, Pilla and Lindsay 1999, Foulley and van Dyk 2000, van Dyk 2000b). Although the PXEM algorithm is a powerful tool for fast stable computation of ML estimates, when prior information is available, PXEM is not very useful. This is because PXEM generally does not result in a closed form M-step even when the corresponding EM algorithm is in closed form (van Dyk 2000b). In this paper, we show how a variant of the one-step-late EM algorithm (Green 1990) can be combined with PXEM to establish a fast closed form algorithm. Our proposal is more stable than Green's one-step-late algorithm in that it guarantees monotone convergence, at least when the corresponding EM algorithm is in closed form.

The main results appear in Section 2, where we show how the EM, PXEM, and one-step-late EM algorithms are combined in the one-step-late PXEM algorithm. In Section 3 we apply the one-step-late PXEM algorithm to compute MAP estimates in probit regression, illustrating the computational gain over EM. A more extended example, applying our methods to compute MAP estimates in the dynamic linear model (DLM), is presented in Section 4 and illustrated numerically in Section 5. Concluding remarks appear in Section 6 and details of the DLM implementation appear in Appendix A.

0960-3174 © 2003 Kluwer Academic Publishers

2. The PXEM and OSL-PXEM algorithms

2.1. *The EM algorithm*

The EM algorithm (Dempster, Laird and Rubin 1997) is an iterative method designed to calculate ML or MAP estimates in models with missing data or latent structure. Suppose, for example, we wish to compute $\hat{\xi} = \text{argmax}_\xi \ell(\xi \mid Y_{\text{obs}})$, where Y_{obs} is the observed data, ξ is a model parameter, and ℓ represents the log posterior distribution. The method of data augmentation embeds the observed-data model into a larger augmented-data model, via a many-to-one mapping, $\mathcal{M}(Y_{\text{aug}}) = Y_{\text{obs}}$, such that

$$\int p(Y_{\text{aug}} \mid \xi)\, dY_{\text{aug}} = p(Y_{\text{obs}} \mid \xi); \tag{1}$$

here and throughout the paper we integrate Y_{aug} over the set $\{Y_{\text{aug}} : \mathcal{M}(Y_{\text{aug}}) = Y_{\text{obs}}\}$.

EM uses the augmented-data model to maximize $\ell(\xi \mid Y_{\text{obs}})$ by computing the sequence

$$\begin{aligned} \xi^{(l+1)} &= \arg\max_\xi Q_{\text{EM}}\big(\xi \bigm| \xi^{(l)}\big) \\ &\equiv \arg\max_\xi E\big\{ \log p(Y_{\text{aug}} \mid \xi) + \log p(\xi) \bigm| Y_{\text{obs}}, \xi^{(l)}\big\}, \end{aligned} \tag{2}$$

for $l = 0, 1, \ldots,$ where $p(\xi)$ represents the prior distribution of ξ. Under certain regularity conditions, this sequence converges to a, perhaps local, maximum of $\ell(\xi \mid Y_{\text{obs}})$ and increases $\ell(\xi \mid Y_{\text{obs}})$ at each iteration (Dempster, Laird and Rubin 1997, Wu 1983). Typically, each iteration involves two steps, the expectation or E-step computes $Q_{\text{EM}}(\xi \mid \xi^{(l)})$ and the maximization or M-step maximizes $Q_{\text{EM}}(\xi \mid \xi^{(l)})$. The global rate of convergence of the algorithm is the smallest eigenvalue of the fraction of observed information, $I_{\text{obs}} I_{\text{aug}}^{-1}$, where I_{obs} is the observed Fisher information and $I_{\text{aug}} = -\frac{\partial^2}{\partial\xi \cdot \partial\xi'} Q(\xi \mid \hat{\xi})|_{\xi=\hat{\xi}}$ is the expected augmented-data Fisher information. In Section 2.2 we discuss recent advances which introduce a creative choice of $p(Y_{\text{aug}} \mid \xi)$ to speed convergence while maintaining easy implementation.

2.2. *Working parameters and the PXEM algorithm*

The working parameter method (Meng and van Dyk 1997) can dramatically improve the convergence rate of EM by taking advantage of the many potential augmented-data models, $p(Y_{\text{aug}} \mid \xi)$, which satisfy (1). In particular, we can parameterize a class of such models using a working parameter, α, such that

$$\int p(Y_{\text{aug}} \mid \xi, \alpha)\, dY_{\text{aug}} = p(Y_{\text{obs}} \mid \xi) \quad \text{for each } \alpha \in \mathcal{A}. \tag{3}$$

Generally, we construct $\mathcal{A}$ such that there exists $\alpha_0 \in \mathcal{A}$ corresponding to the "standard data augmentation" scheme, i.e., $p(Y_{\text{aug}} \mid \xi, \alpha_0) = p(Y_{\text{aug}} \mid \xi)$. Meng and van Dyk (1997) show how α can be chosen to optimize the rate of convergence of the resulting EM algorithm by minimizing the expected augmented-data Fisher information as a function of α. The method of efficient augmentation implements the EM algorithm with the working parameter fixed at this optimal value; see Meng and van Dyk (1998) and van Dyk (2000a) for applications of this method.

As a variant of the working parameter method, Liu, Rubin and Wu (1998) suggest fitting α during the iteration rather than optimizing over α before running the algorithm. In particular, at iteration l, the E-step of the PXEM algorithm computes

$$\begin{aligned} Q_{\text{PXEM}}\big(\xi, \alpha \bigm| \xi^{(l)}, \alpha_0\big) = \int &\{\log p(Y_{\text{aug}} \mid \xi, \alpha) + \log p(\xi)\} \\ &\times p\big(Y_{\text{aug}} \bigm| Y_{\text{obs}}, \xi^{(l)}, \alpha_0\big)\, dY_{\text{aug}}, \end{aligned}$$

and the M-step sets $(\xi^{(l+1)}, \alpha^{(l+1)}) \equiv \text{argmax}_{(\xi,\alpha)} Q_{\text{PXEM}}(\xi, \alpha \mid \xi^{(l)}, \alpha_0)$; $\alpha^{(l+1)}$ is not used subsequently. Like the EM algorithm, PXEM increases the loglikelihood at each iteration (Liu, Rubin and Wu 1998, van Dyk 2000b), but PXEM's global rate of convergence is at least as good as that of the algorithm that fixes α at α_0, i.e., the standard EM algorithm. Heuristically, conditioning on $\alpha = \alpha_0$ increases the augmented-data information. By eliminating this conditioning, the augmented-data information decreases and the rate of convergence is improved. As pointed out by Liu, Rubin and Wu (1998), the larger the dimension of α, the larger the potential gain of PXEM; see also Foulley and van Dyk (2000), van Dyk and Meng (2000).

2.3. *Extending PXEM for Bayesian calculations*

Although Liu, Rubin and Wu (1998) illustrated the dramatic improvement of PXEM over EM for computing ML estimates in a number of examples (see also van Dyk 2000b), PXEM is not generally useful for computing MAP estimates or penalized ML estimates even in cases when the corresponding Bayesian EM algorithm can be easily implemented. To identify the difficulty with Bayesian calculations and to illustrate the working parameter method and PXEM, we consider the simple random-effects model

$$y_i \mid \theta_i \overset{\text{indep}}{\sim} N(\theta_i, \sigma^2) \text{ with } \theta_i \sim N(0, \tau^2) \text{ and } \tau^2 \overset{\text{iid}}{\sim} \frac{\eta\tau_0^2}{\chi_\eta^2}, \tag{4}$$

for $i = 1, \ldots, n$, where η and τ_0^2 are fixed hyperparameters. For simplicity, we also assume σ^2 is fixed. A simple EM algorithm defines $Y_{\text{aug}} = \{y = (y_1, \ldots, y_n), \theta = (\theta_1, \ldots, \theta_n)\}$ and can be extended to a PXEM algorithm using $Y_{\text{aug}} = \{y, \phi = (\phi_1, \ldots, \phi_n)\}$, where $\phi_i = \theta_i/\alpha$ and α is the working parameter. Thus, the model can be rewritten as

$$y_i \mid \phi_i \sim N(\alpha\phi_i, \sigma^2) \text{ with } \phi_i \sim N(0, \tau^2/\alpha^2) \text{ and } \tau^2 \sim \frac{\eta\tau_0^2}{\chi_\eta^2},$$

which yields the same marginal distribution, $p(y \mid \tau^2)$, as (4), for any $\alpha \neq 0$. With $\alpha_0 = 1$, we can write

$$
\begin{aligned}
&Q_{\mathrm{PXEM}}\{\tau^2, \alpha \mid (\tau^2)^{(l)}, \alpha_0\}\\
&\quad = E\left\{ -\sum_{i=1}^{n} \frac{(y_i - \alpha\phi_i)^2}{2\sigma^2} - \frac{\alpha^2\phi_i^2}{2\tau^2} \,\middle|\, y, (\tau^2)^{(l)}, \alpha_0 \right\}\\
&\qquad - \frac{n}{2}\log\left(\frac{\tau^2}{\alpha^2}\right) - \left(\frac{\eta}{2} + 1\right)\log\tau^2 - \frac{\eta\tau_0^2}{2\tau^2}.
\end{aligned}
$$

For ML calculations, we set $\eta = -2$ and $\tau_0^2 = 0$ to induce a flat prior distribution. We can then introduce the reparameterization $(\tilde{\tau}^2, \alpha)$ with $\tilde{\tau}^2 = \tau^2/\alpha^2$ in order to obtain a closed form M-step,

$$(\tilde{\tau}^2)^{(l+1)} = \sum_{i=1}^{n} E\{\theta_i^2 \mid y, (\tau^2)^{(l)}\}, \tag{5a}$$

$$\alpha^{(l+1)} = \frac{\sum_{i=1}^{n} y_i E\{\theta_i \mid y, (\tau^2)^{(l)}\}}{\sum_{i=1}^{n} E\{\theta_i^2 \mid y, (\tau^2)^{(l)}\}}, \tag{5b}$$

and $(\tau^2)^{(l+1)} = (\alpha^{(l+1)})^2(\tilde{\tau}^2)^{(l+1)}$. The expectations in (5) are standard Gaussian calculations and are computed in the E-step; here we use the simplification $E\{\phi_i \mid y, (\tau^2)^{(l)}, \alpha = \alpha_0\} = E\{\theta_i \mid y, (\tau^2)^{(l)}\}$ and likewise for the expectation of ϕ_i^2. Reparametrizing the model parameter, ξ, is the standard method for obtaining a simple M-step in PXEM. Unfortunately, we cannot find a closed-form M-step using this method when a prior distribution is used; e.g., if we use other values of (η, τ_0^2) in this example.

This difficulty was noted by van Dyk (2000b) who suggested using the efficient augmentation when PXEM is difficult to implement. Here, we adopt a different strategy, a variant of the one-step-late method of Green (1990). We consider the typical case where the M-step of a PXEM algorithm for ML solves the score function via an invertible transformation $(\tilde{\xi}, \alpha) = g(\xi, \alpha)$; in our example, this corresponds to $(\tilde{\tau}^2, \alpha) = g(\tau^2, \alpha)$. In the one-step-late PXEM algorithm, which computes a MAP estimate, we set $(\xi^{(l+1)}, \alpha^{(l+1)}) = g^{-1}(\tilde{\xi}^{(l+1)}, \alpha^{(l+1)})$, where $(\tilde{\xi}^{(l+1)}, \alpha^{(l+1)})$ approximately maximizes

$$
\begin{aligned}
Q_{\mathrm{PXEM}}\big(\xi, \alpha \mid \xi^{(l)}, \alpha_0\big) = E\big\{ \log p(Y_{\mathrm{aug}} \mid \tilde{\xi}, \alpha) \bigm| Y_{\mathrm{obs}}, \xi^{(l)}, \alpha_0 \big\}&\\
+ \log p(g^{-1}(\tilde{\xi}, \alpha));&
\end{aligned}
\tag{6}
$$

the distribution in the second term is $p(\xi)$, the prior on ξ, which depends on (ξ, α) only through ξ. To approximately maximize (6), we set $(\tilde{\xi}^{(l+1)}, \alpha^{(l+1)})$ to the solution of

$$\frac{\partial}{\partial(\tilde{\xi}, \alpha)} E\big\{ \log p(Y_{\mathrm{aug}} \mid \tilde{\xi}, \alpha) \bigm| Y_{\mathrm{obs}}, \xi^{(l)}, \alpha_0 \big\} \tag{7a}$$

$$+ \frac{\partial}{\partial(\tilde{\xi}, \alpha)} \log p(g^{-1}(\tilde{\xi}, \alpha))\bigg|_{\alpha = \alpha_0} = 0. \tag{7b}$$

This is a one-step-late algorithm since we fix α at its input value in the reparametrized prior distribution for ξ; coincidentally, the input value is also the convergent value. Since $(\hat{\xi}, \alpha_0)$ is a fixed point of PXEM (see Liu, Rubin and Wu 1998), it is also a fixed point of this algorithm. This procedure is generally in closed form when the corresponding EM algorithm is in closed form, i.e., with suitable choice of conjugate prior distribution. If this is not the case, it may be helpful to evaluate (7b) at $(\tilde{\xi}, \alpha) = g(\xi^{(l)}, \alpha_0)$ so that only (7a) depends on $(\tilde{\xi}, \alpha)$. In some cases, we evaluate only certain factors of (7b) at α_0. Generally some creativity is required to derive useful algorithms; this is in the spirit of other EM-type algorithms, which rely on a creative choice of Y_{aug}. The emphasis is on arriving at a simple M-step; several examples appear in Sections 3 and 4.

Unfortunately, this one-step-late algorithm is not guaranteed to increase the log posterior distribution at each iteration (see Green 1990). However, it is often easy to replace the one-step-late update with the update from EM if the log posterior distribution decreases, thus guaranteeing monotone convergence. We define the one-step-late PXEM algorithm in this way; at iteration l, this algorithm proceeds as follows:

E-step: Compute $Q_{\mathrm{PXEM}}(\xi, \alpha \mid \xi^{(l)}, \alpha_0)$;
M-step: Set $(\xi_{\mathrm{prop}}^{(l+1)}, \alpha_{\mathrm{prop}}^{(l+1)}) = g^{-1}(\tilde{\xi}_{\mathrm{prop}}^{(l+1)}, \alpha_{\mathrm{prop}}^{(l+1)})$ where $(\tilde{\xi}_{\mathrm{prop}}^{(l+1)}, \alpha_{\mathrm{prop}}^{(l+1)})$ solves (7b);
Correction-step: If $l(\xi_{\mathrm{prop}}^{(l+1)} \mid Y_{\mathrm{obs}}) \leq l(\xi^{(l)} \mid Y_{\mathrm{obs}})$, set $\xi^{(l+1)} = \arg\max_{\xi} Q_{\mathrm{EM}}(\xi \mid \xi^{(l)})$, otherwise set $\xi^{(l+1)} = \xi_{\mathrm{prop}}^{(l+1)}$.

In the correction step, it is often the case that $\arg\max_{\xi} Q_{\mathrm{EM}}(\xi \mid \xi^{(l)}) = \tilde{\xi}_{\mathrm{prop}}^{(l+1)}$, a quantity that has already been computed. Thus, this algorithm results in a simple monotone adaptation of PXEM for computing posterior modes and is generally quite efficient, especially in the important case when the prior distribution is weak relative to the likelihood. Generally, we find that, as with the ordinary PXEM algorithm, this algorithm offers the most gain when EM is the slowest to converge, see Liu, Rubin and Wu (1998). In the remainder of the paper, we refer to this algorithm as the OSL-PXEM algorithm.

We illustrate the OSL-PXEM algorithm by fitting model (4). The E-step is exactly as in EM and PXEM for ML: we compute $E\{\theta_i \mid y, (\tau^2)^{(l)}\}$ and $E\{\theta_i^2 \mid y, (\tau^2)^{(l)}\}$. In the M-step, we solve (7) via

$$
\begin{aligned}
(\tilde{\tau}^2)_{\mathrm{prop}}^{(l+1)} &= \frac{\sum_{i=1}^{n} E\{\theta_i^2 \mid y, (\tau^2)^{(l)}\} + \eta\tau_0^2}{n + \eta + 2},\\
\alpha_{\mathrm{prop}}^{(l+1)}&\\
&= \frac{\sum_{i=1}^{n} y_i E\{\theta_i \mid y, (\tau^2)^{(l)}\} - \sigma^2\{(\eta + 2) - \eta\tau_0^2/(\tilde{\tau}^2)_{\mathrm{prop}}^{(l+1)}\}}{\sum_{i=1}^{n} E\{\theta_i^2 \mid y, (\tau^2)^{(l)}\}},
\end{aligned}
$$

and set $(\tau^2)_{\mathrm{prop}}^{(l+1)} = (\alpha_{\mathrm{prop}}^{(l+1)})^2(\tilde{\tau}^2)_{\mathrm{prop}}^{(l+1)}$.

To illustrate the efficiency of this strategy, 1500 data sets, each of size 1000 were generated according to model (4) with $\tau^2 = 1$. The value of σ^2 was set to one of ten values (0.25, 1, 4, 9, 16, 25, 36, 49, 64, 81) in equal proportion. Model (4) was fit using both EM and OSL-PXEM with one of three prior distributions on τ^2, $p(\tau^2) \propto 1$, $\tau^2 \sim 0.5/\chi_1^2$, and $\tau^2 \sim 10/\chi_{20}^2$, again in equal proportion. With a flat prior distribution (i.e., ML fitting), the

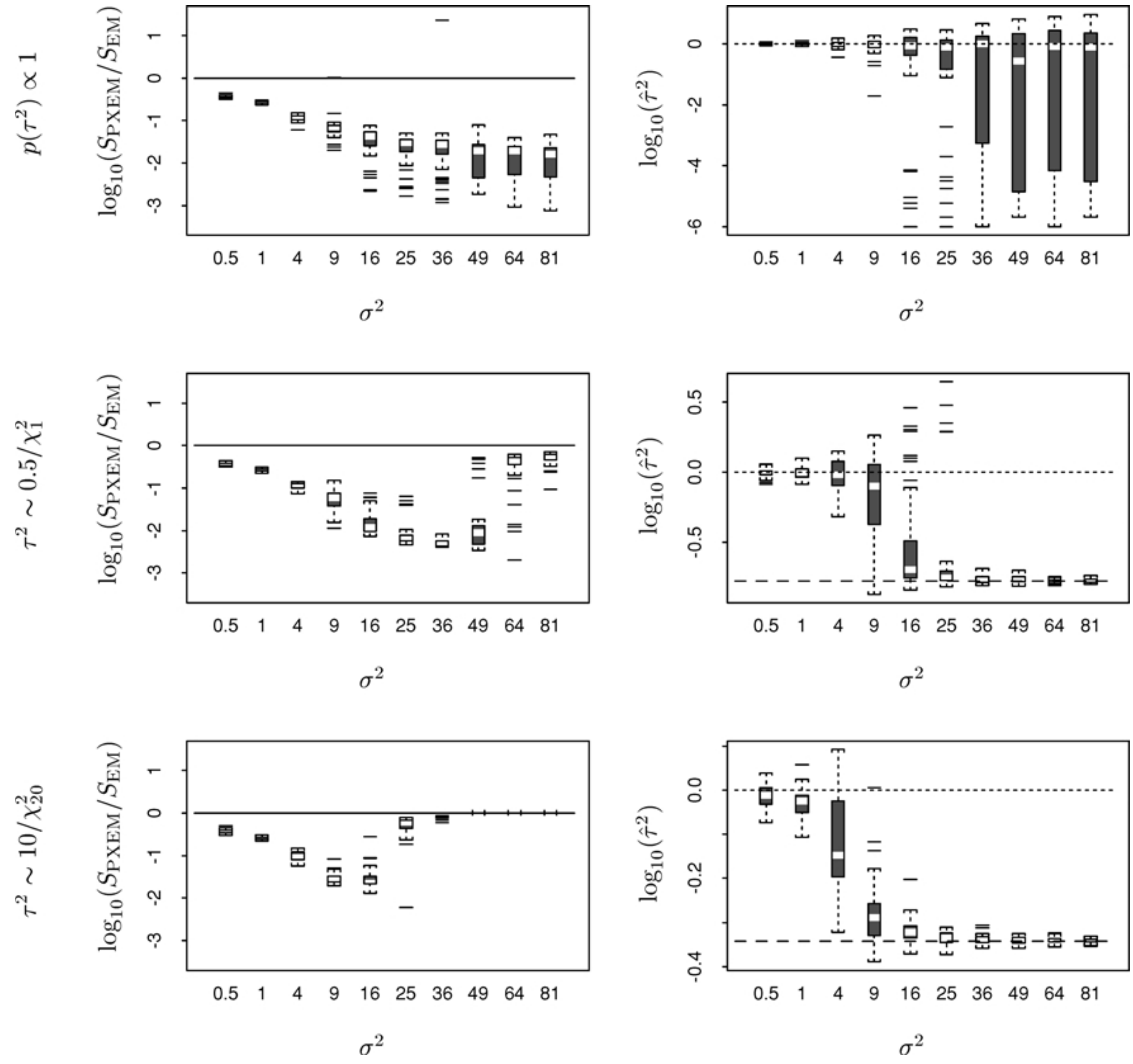

Fig. 1. *The computational gain of OSL-PXEM over EM when fitting a random-effects model. The boxplots illustrate the efficiency of OSL-PXEM for fitting model (4) with various values of σ^2. The first column plots $\log_{10}$ (S_{PXEM}/S_{EM}), the computational time required by OSL-PXEM in units of that required by EM and the second column plots the fitted value of τ^2, also on the $\log_{10}$ scale. The rows correspond to three prior distributions, $p(\tau^2) \propto 1$, $\tau^2 \sim 0.5/\chi_1^2$, and $\tau^2 \sim 10/\chi_{20}^2$. In the second column, the dotted lines represent the true value of τ^2 and the dashed lines the prior mode of τ^2. The gain offered by OSL-PXEM grows with σ^2 but is dampened as the prior distribution becomes strong relative to the likelihood*

one-step-late method was not required and we used the standard PXEM algorithm. The value of σ^2 was assumed known in model fitting. Each algorithm was started at $(\tau^2)^{(0)} = 1$ and run until the log posterior distribution increased by less than 10^{-7}. The results of the simulation appear in Fig. 1, where the rows correspond to the three prior distributions. The box plots in the first column show the log (base 10) of the time required by OSL-PXEM in units of the time required by EM to fit the same model to the same data set. Notice that for ML fitting with large values of σ^2, EM took about 100 times longer than PXEM. A similar trend can be seen with proper prior distributions except the advantage of OSL-PXEM vanishes for large values of σ^2. The second column of Fig. 1 explains this phenomenon by plotting the fitted values of τ^2 on the $\log_{10}$ scale. Here the dotted line represent the value of τ^2 used to generate the data and the dashed line the prior mode. As σ^2 grows, the prior distribution becomes more influential. When the prior specification swamps the likelihood, OSL-PXEM offers little or no advantage over EM.

3. Computing the MAP estimate in probit regression

3.1. *Probit regression model*

In this section, we describe a fast new OSL-PXEM algorithm for computing MAP estimates of the coefficients in probit regression. Probit regression is a generalized linear model for

Bernoulli data, which uses a probit link function. Specifically, we assume

$$Y_i \mid \xi \overset{\text{indep}}{\sim} \text{Bernoulli}\{\Phi(X_i'\xi)\}, \quad \text{for } i = 1, \ldots, n, \tag{8}$$

where Y_i is an observed binary response, X_i is a $(p \times 1)$ vector of observed covariates, ξ is a $(p \times 1)$ vector of coefficients, and Φ is the cumulative distribution function of the standard normal distribution. Model (8) is routinely fit with a proper prior distribution on ξ, for example, in economics (McCulloch and Rossi 1994, Imai and van Dyk 2003).

A well-known data-augmentation scheme that is useful for fitting (8) using EM can be written

$$Y_i = \text{sign}(Z_i) \quad \text{with } Z_i \mid \xi \sim N(X_i'\xi, 1), \tag{9}$$

for $i = 1, \ldots, n$, where $Z = (Z_1, \ldots, Z_n)'$ are unobserved latent Gaussian responses and are treated as missing data (see, e.g., Albert and Chib 1993). Liu, Rubin and Wu (1998) describe the standard EM algorithm based on (9) and a much faster PXEM algorithm for computing the ML estimate for model (8). An advantage of these EM-type algorithms over other methods (e.g., iterative reweighted least squares) for probit regression is that they are easily generalized to accommodate missing data or other model components such as random effects. EM-type algorithms can also be used as a starting point for constructing a Gibbs sampler for posterior simulation and exhibit more stable convergence when computing posterior modes. Here we generalize the standard EM algorithm and derive a new faster OSL-PXEM; both algorithms are designed to account for proper prior information for ξ. In particular, we allow for the standard multivariate normal (semi-conjugate) prior distribution, $\xi \sim N(\xi_0, T)$. We conclude with an example that demonstrates the superior convergence properties of the OSL-PXEM algorithm.

3.2. *An EM algorithm*

An EM iteration for computing a MAP estimate is given by (2). For probit regression, using $Y_{\text{aug}} = (Y, Z)$ with $Y = (Y_1, \ldots, Y_n)'$, this iteration takes the form

$$\xi^{(l+1)} = \arg\max_{\xi} E\left\{-\frac{1}{2}\sum_i (Z_i - X_i'\xi)^2 - \frac{1}{2}(\xi - \xi_0)'T^{-1}(\xi - \xi_0) \,\middle|\, Y, \xi^{(l)}\right\}, \tag{10}$$

which we compute using the familiar two-step formulation:

E-step: Compute the expected value in (10) which simplifies to computing the expected value of each Z_i under the truncated normal distribution, $p(Z_i \mid Y_i, \xi^{(l)})$,

$$Z_i^{(l+1)} = E\big(Z_i \,\big|\, Y, \xi^{(l)}\big) = \begin{cases} X_i'\xi^{(l)} + \phi\big(X_i'\xi^{(l)}\big)/\big\{1 - \Phi\big(-X_i'\xi^{(l)}\big)\big\} & \text{if } Y_i = 1 \\ X_i'\xi^{(l)} - \phi\big(X_i'\xi^{(l)}\big)/\big\{- \Phi\big(-X_i'\xi^{(l)}\big)\big\} & \text{if } Y_i = 0, \end{cases}$$

where ϕ is the probability density function of the standard normal distribution. (Here we take advantage of the fact that as a function of ξ, the augmented-data log posterior is linear in Z.)

M-step: Compute the maximization in (10), $\xi^{(l+1)} = (X'X + T^{-1})^{-1}(X'Z^{(l+1)} + T^{-1}\xi_0)$, where $Z^{(l+1)} = (Z_1^{(l+1)}, \ldots, Z_n^{(l+1)})'$.

We use the SWEEP operator (as discussed in Little and Rubin 1987, pages 53–57) to accomplish the matrix inversion in the M-step. In particular, we sweep out the first p columns and rows of the matrix

$$\begin{pmatrix} \sum_{i=1}^n X_i X_i' + T^{-1} & \sum_{i=1}^n X_i' Z_i^{(l+1)} + T^{-1}\xi_0 \\ \sum_{i=1}^n X_i Z_i^{(l+1)} + \xi_0' T^{-1} & \gamma + \xi_0' T^{-1}\xi_0 \end{pmatrix} \tag{11}$$

and obtain $\xi^{(l+1)}$ as the upper right $(p \times 1)$ submatrix of the result. Since we are only interested in the regression coefficients, we may set γ to any value.

3.3. *An OSL-PXEM algorithm*

We give a more detailed description of the OSL-PXEM algorithm. We begin by introducing the working parameter as in Liu, Rubin and Wu (1998), by setting $\tilde{Z}_i = \alpha Z_i$ for each i. The augmented-data model is

$$Y_i = \text{sign}(\tilde{Z}_i) \quad \text{with } \tilde{Z}_i \mid (\tilde{\xi}, \alpha) \sim N(X_i'\tilde{\xi}, \alpha^2), \tag{12}$$

for $i = 1, \ldots, n$, where $\tilde{\xi} = \alpha\xi$. The marginal distribution of the observed data implied by the joint distribution on $(Y, \tilde{Z})$ given in (12) is $Y_i \mid (\tilde{\xi}, \alpha) \overset{\text{indep}}{\sim} \text{Bernoulli}\{\Phi(X_i'\tilde{\xi}/\alpha)\}$. Since this is the same as is implied by the joint distribution given in (9), condition (3) is satisfied.

The OSL-PXEM algorithm consists of the three steps given in Section 2.3. In the E-step, we compute $Q_{\text{PXEM}}(\xi, \alpha \mid \xi^{(l)}, \alpha_0)$ as

$$-\frac{n}{2}\log\alpha^2 - \frac{1}{2\alpha^2}\sum_{i=1}^n \{B_i^{(l+1)} - 2Z_i^{(l+1)}X_i'\tilde{\xi} + \tilde{\xi}'(X_i X_i')\tilde{\xi}\} - \frac{\xi_0' T^{-1}\xi_0}{2} + \frac{\tilde{\xi}' T^{-1}\xi_0}{\alpha} - \frac{\tilde{\xi}' T^{-1}\tilde{\xi}}{2\alpha^2}, \tag{13}$$

where $B_i^{(l+1)} = E(\tilde{Z}_i^2 \mid Y, \xi^{(l)}, \alpha_0)$ and $Z_i^{(l+1)} = E(\tilde{Z}_i \mid Y, \xi^{(l)}, \alpha_0)$. Because of the working parameter, the augmented-data log posterior is no longer linear in Z and we are required to compute $B_i^{(l+1)}$ for each i.

In the M-step, we approximately maximize (13) by solving (7). To this end, we differentiate (13) with respect to $\tilde{\xi}$ and α, yielding

$$\frac{\partial Q}{\partial \tilde{\xi}} = -\frac{1}{\alpha^2}\sum_{i=1}^n \{-Z_i^{(l+1)}X_i + (X_i X_i')\tilde{\xi}\} - \frac{1}{\alpha^2}(-\alpha T^{-1}\xi_0 + T^{-1}\tilde{\xi}) \tag{14}$$

and

$$\frac{\partial Q}{\partial \alpha^2} = -\frac{n}{2\alpha^2} + \frac{1}{2(\alpha^2)^2}\sum_{i=1}^{n}\left\{B_i^{(l+1)} - 2Z_i^{(l+1)}X_i'\tilde{\xi} + \tilde{\xi}'(X_i'X_i)\tilde{\xi}\right\} - \frac{\tilde{\xi}'T^{-1}\xi_0}{2(\alpha^2)^{3/2}} + \frac{\tilde{\xi}'T^{-1}\tilde{\xi}}{2(\alpha^2)^2}. \quad (15)$$

We compute $\tilde{\xi}_{\text{prop}}^{(l+1)}$ by setting $\alpha = \alpha_0 = 1$ in the numerator of (14), setting the result equal to zero, and solving. This is a slight adaptation of (7), which specifies that α be set to α_0 in both the numerator and denominator of (14). This adaptation is in the spirit of the one-step-late strategy and simplifies calculations in this problem. The result is that we compute $\tilde{\xi}_{\text{prop}}^{(l+1)}$ exactly as $\xi^{(l+1)}$ was computed in the EM algorithm. To compute $\alpha^{(l+1)}$ we replace $\tilde{\xi}$ with $\tilde{\xi}_{\text{prop}}^{(l+1)}$ in $\partial Q/\partial\alpha^2$, set the denominator of the first term of (15) equal to $\alpha^2(\alpha_0^2)^{1/2}$, set the resulting score function equal to zero, and solve. Again we are not setting all the factors of $\alpha = \alpha_0$ in the two terms in (15), but as few as are necessary for simple computation.

The OSL-PXEM iteration is completed with the correction step:

E-step: For each i, compute $Z_i^{(l+1)}$ exactly as in the EM algorithm and

$$B_i^{(l+1)} = E\left(\tilde{Z}_i^2 \mid Y_{\text{obs}}, \xi^{(l)}, \alpha_0\right) = \left(X_i'\xi^{(l)}\right)^2 + 1 + X_i'\xi^{(l)}\left(Z_i^{(l+1)} - X_i'\xi^{(l)}\right).$$

M-step: First we obtain, $\tilde{\xi}_{\text{prop}}^{(l+1)} = (X'X + T^{-1})^{-1}(X'Z^{(l+1)} + T^{-1}\xi_0)$, using the SWEEP operator as described in Section 3.2, but this time with γ replaced by $\sum_{i=1}^{n} B_i^{(l+1)}$. We set $(s^2)^{(l+1)}$ equal to the lower right (1×1) submatrix of (11) after sweeping, i.e.,

$$(s^2)^{(l+1)} = \sum_{i=1}^{n}\left\{B_i^{(l+1)} - 2Z_i^{(l+1)}X_i'\tilde{\xi}_{\text{prop}}^{(l+1)} + \left(\tilde{\xi}_{\text{prop}}^{(l+1)}\right)'(X_i'X_i)\tilde{\xi}_{\text{prop}}^{(l+1)}\right\} + \xi_0'T^{-1}\xi_0 - 2\left(\tilde{\xi}_{\text{prop}}^{(l+1)}\right)'T^{-1}\xi_0 + \left(\tilde{\xi}_{\text{prop}}^{(l+1)}\right)'T^{-1}\tilde{\xi}_{\text{prop}}^{(l+1)}.$$

And finally set

$$(\alpha^2)^{(l+1)} = \frac{(s^2)^{(l+1)} - \xi_0'T^{-1}\xi_0 + 2\left(\tilde{\xi}_{\text{prop}}^{(l+1)}\right)'T^{-1}\xi_0}{n + \left(\tilde{\xi}_{\text{prop}}^{(l+1)}\right)'T^{-1}\xi_0}$$

and $\xi_{\text{prop}}^{(l+1)} = \tilde{\xi}_{\text{prop}}^{(l+1)}/\alpha^{(l+1)}$.

Correction-step: If $\ell(\xi_{\text{prop}}^{(l+1)} \mid Y) > \ell(\xi^{(l)} \mid Y)$, set $\xi^{(l+1)} = \xi_{\text{prop}}^{(l+1)}$; otherwise, set $\xi^{(l+1)} = \tilde{\xi}_{\text{prop}}^{(l+1)}$.

Next, we illustrate the computational performance of this algorithm.

Table 1. *Component forms for the dynamic linear model*

Component Form	F_t	G
Polynomial Trend (or Growth*) of Order p	$\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{p\times 1}$	$\begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}_{p\times p}$
Seasonal Effect with $p+1$ Seasons†	$\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{(p+1)\times 1}$	$\begin{pmatrix} 0 & I_p \\ 1 & 0' \end{pmatrix}_{(p+1)\times(p+1)}$
Harmonic Component	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} \cos\omega & \sin\omega \\ -\sin\omega & \cos\omega \end{pmatrix}$
Regression Component	$(X_t)_{p\times 1}$	I_p

*The polynomial growth and trend components differ in their constraints on the system variance, T; see Appendixes A.1 and A.2.

†We constrain $1'_{p+1}\theta = 0$, therefore, $T1_{p+1} = 0$; 1_p is a $(p \times 1)$ vector of ones and I_p is the $(p \times p)$ identity matrix. Alternatively, we can rewrite F_t' and G as described in Appendix A.3.

3.4. *Computational performance*

To compare the performance of the standard EM and OSL-PXEM algorithms we implemented both algorithms using a data set supplied by M. Haas, who was a client of the statistics consulting program at the University of Chicago. The data consists of two covariates that are used to predict the occurrence of latent membranous lupus nephritis. The data consists of measurement of 55 patients 18 of whom have been diagnosed with latent membranous lupus. (See Haas (1994, 1998) for scientific background; the data appear in Table 1 of van Dyk and Meng (2001).)

We compute MAP estimates using a variety of prior distributions to investigate the relative efficiency of EM and OSL-PXEM. Using a normal prior distribution, we set the prior mean of ξ to zero and the prior variance $T = cI$, varying $\log_{10} c$ between -1 and 3. Each of the models was fit using both the EM and the OSL-PXEM algorithm with the same starting values and convergence criterion. The relative efficiency of the algorithms is summarized in Fig. 2. We plot $\log_{10}(S_{\text{OSL-PXEM}}/S_{\text{EM}})$, the log (base 10) of the ratio of time (in seconds) required by OSL-PXEM to the time required by EM, against c. Larger values of c indicates weaker influence of the prior distribution on the posterior distribution, as is illustrated in the middle plot where we compared the MAP estimates with prior mode and ML estimates for each c. We also compared the relative time with $\log_{10} S_{\text{EM}}$. It is clear from the plots that when the prior distribution is dominant, MAP estimates are close to the prior mode, and OSL-PXEM offers little gain over EM. However, when c is larger, the data become more influential, EM becomes much slower to converge, and OSL-PXEM becomes faster relative to EM. When c is large (i.e., 10^3), the posterior mode is essentially

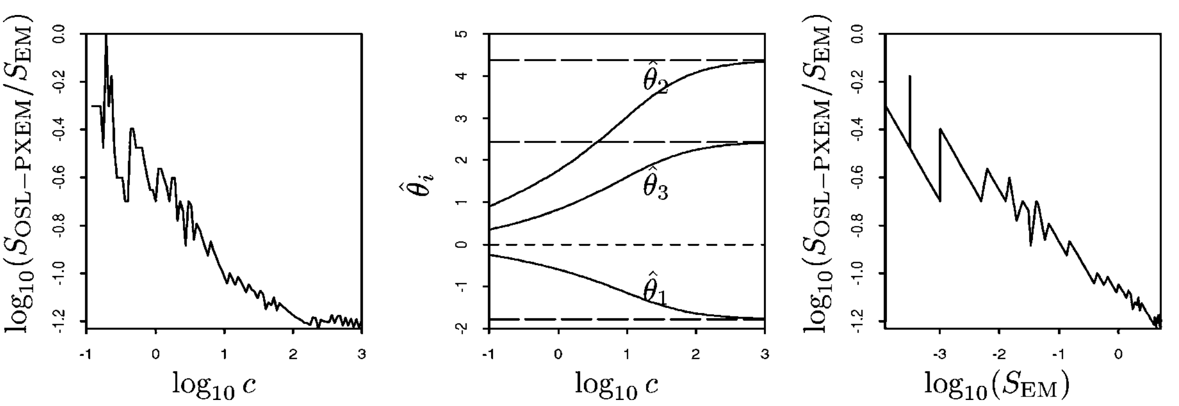

Fig. 2. *Computational gain of OSL-PXEM when fitting a probit regression model. The figure plots the relative time required by OSL-PXEM and EM (log base 10 scale) for fitting the probit regression model to the kidney data, against* $\log_{10}c$ *(first plot) and against* $\log_{10}(\tau_{EM})$ *(final plot). The middle plot shows how the MAP estimates of* θ *are affected by the prior variance; the solid lines are the MAP estimates; the long dashed lines are the ML estimates; and the short dashed line is prior mode. When c is small, the MAP estimates are close to the prior mode and OSL-PXEM offers little gain in computational time. However, when c is large, the MAP estimates are close to the ML estimate, and OSL-PXEM saves as much as 94% of the computational time. Fortunately, OSL-PXEM offers the greatest gain when EM is the slowest to converge*

the ML estimate and OSL-PXEM offers the biggest gain, saving about 94% of the computational time required by EM.

4. Computing the MAP estimate in the DLM

4.1. *The dynamic linear model*

The class of structural time series models known alternatively as state-space models and dynamic linear models, or DLMs, are enormously powerful for inference and forecasting in time-dependent dynamic systems. Introduced in the 1960s (e.g., Harrison 1965, 1967), today these models are popular in a wide range of applications including population dynamics, demographics, agriculture, and economics; see Harvey (1989) for a frequentist perspective and West and Harrison (1997) and Pole, West and Harrison (1994) for a Bayesian perspective. We consider a class of Gaussian DLMs, which hypothesizes an underlying evolution in time of the coefficients of a Gaussian linear model, i.e.,

$$\begin{aligned} &\text{Observation equation: } y_t = X_t'\beta + F_t'\theta_t + \nu_t, && \nu_t \sim N(0,\sigma^2); \qquad (16)\\ &\text{System equation: } \theta_t = G\theta_{t-1} + \omega_t, && \omega_t \sim N(0,T);\\ &\text{Initial condition: } \theta_0 \sim N(0,\kappa T), \end{aligned}$$

where $t = 1, 2, \ldots, n$, $y = (y_1, \ldots, y_n)'$ is an observed response variable, X_t $(q \times 1)$, F_t $(p \times 1)$ and G $(p \times p)$ are known coefficients, β is a $(q \times 1)$ fixed effect, and θ_t is a $(p \times 1)$ parameter evolving over time. In model (16), the system equation represents a first order Markov process for the evolution of θ_t, which in turn acts as a regression parameter in the observation equation. The scalar σ^2 represents the level-one variance and T the level-two $(p \times p)$ variance-covariance matrix. The scalar κ is a known constant, typically greater than one, which reflects uncertainty in the initial condition. The zero mean in the initial condition is assumed without loss of generality, since if we assume an initial mean of μ, model (16) is recovered by replacing y_t and θ_t with $y_t - F_t'G^t\mu$ and $\theta_t - G^t\mu$ respectively, where the superscript t represents a power. A model including a fixed effect in the system equation can also be written in the form of model (16) by transforming y_t and θ_t.

Although models of this form go under various names, we adopt the standard notation and terminology of West and Harrison (1997) in the context of the DLM. Table 1 describes several standard forms of the DLM, which West and Harrison use to describe polynomial and cyclic trends and to incorporate covariate information. These so-called component forms can be combined by introducing a block structure in G and T to create a flexible class of models, which for example, can accommodate a seasonal effect around a linear growth model while correcting for various covariates.

Various extensions to model (16) are possible and useful in practice. For example, either F or G may be an unknown model parameter and the observed response may be multivariate. Such generalizations might be handled using EM-type algorithms and the working parameter methods described here may help improve convergence properties. We avoid these issues in this article, however, because they require creative solutions and more complicated algorithms (e.g., with the M-step replaced by a set of conditional M-steps) which would take us afield from our goal of illustrating OSL-PXEM.

The popularity of DLMs has been fostered by a number of computational tools such as the BATS package (Pole, West and Harrison 1994) which uses the method of variance discounting to estimate model parameters. The Kalman filter (Kalman 1960, Kalman and Bucy 1961) and the forward-backward algorithm (Jazwinski 1970) both aim to estimate the underlying trend given all other model parameters. ML estimation of these parameters can be accomplished via scoring, Newton-Raphson techniques (e.g., Gupta and Mehra 1974, Ledolter 1979, Goodrich and Caines 1979), or the EM algorithm (Shumway and Stoffer 1982, Watson and Engle 1983). Although much work has been devoted to methods for fitting the DLM, we know of no off-the-shelf software for computing MAP estimates.

In the remainder of Section 4, we focus on new methods for computing MAP estimates. The relative size of the error variance of the observation and system equations controls the system volatility and thus the predictive power of the model. Unfortunately, these parameters can be very difficult to estimate with precision and prior information when available can be critical for forecasting. Thus, we focus on new efficient algorithms for computing MAP estimates.

4.2. *Shumway and Stoffer's EM algorithm*

Shumway and Stoffer (1982) first recognized model (16) as a latent variable model which could routinely be fit using the EM algorithm. In particular, they noted that given $Y_{\text{aug}} = \{\theta_0, (\theta_t, y_t), t = 1, \ldots, n\}$, computing the ML estimate of $\xi = (\beta, \sigma^2, T)$ requires only the evaluation of two simple sums of squares, i.e., the M-step of EM is simple. Secondly, given ξ, the Kalman filter can be used in the forward-backward algorithm to easily accomplish the required expectation in the E-step. Iterating these two steps produces a sequence converging to $\hat{\xi}$, the desired maximizer of (1). We give the details of Shumway and Stoffer's algorithm, partially as a means of introducing the notation for our algorithms.

The augmented-data loglikelihood for Shumway and Stoffer's algorithm can be written

$$Q_{\text{EM}}(\xi \mid \xi^{(l)})$$
$$= -\frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{t=1}^{n}\{(y_t - X_t'\beta - F_t'h_t)^2 + F_t'H_tF_t\}$$
$$-\frac{n+1}{2}\log|T| - \frac{1}{2}\text{tr}(T^{-1}V),$$

where

$$V = D - CG' - GC' + GBG' + (H_0 + h_0h_0')/\kappa,$$
$$B = \sum_{t=1}^{n}(H_{t-1} + h_{t-1}h_{t-1}'), \quad C = \sum_{t=1}^{n}(P_t + h_th_{t-1}'),$$
$$D = \sum_{t=1}^{n}(H_t + h_th_t'),$$

$h_t = E(\theta_t \mid y, \xi^{(l)})$, $H_t = \text{var}(\theta_t \mid y, \xi^{(l)})$, and $P_t = \text{cov}(\theta_t, \theta_{t-1} \mid y, \xi^{(l)})$ are all obtained as functions of $\xi^{(l)}$ via the forward-backward algorithm in the E-step; we sometimes write $h_t(\xi^{(l)})$, $H_t(\xi^{(l)})$, $B(\xi^{(l)})$, etc., for clarity. The parameter, ξ, is updated by maximizing $Q_{\text{EM}}(\xi \mid \xi^{(l)})$, i.e., the M-step consists of

$$(\sigma^2)^{(l+1)} = \frac{1}{n}\sum_{t=1}^{n}\{(y_t - F_t'h_t)^2 + F_t'H_tF_t\},$$
$$\beta^{(l+1)} = \left(\sum_{t=1}^{n}X_tX_t'\right)^{-1}\sum_{t=1}^{n}X_t(y_t - F_t'h_t),$$

and

$$T^{(l+1)} = \frac{V}{n+1}. \tag{17}$$

(Shumway and Stoffer's algorithm did not include β.) The final expression assumes T is unconstrained; for some component forms a modification is required to accommodate restrictions; see Appendix A.

4.3. *Adding a working parameter to the model*

To develop a fast algorithm, we begin by introducing a working parameter to index a family of augmented-data models via a simple transformation of the latent variable. In particular, we introduce a scale transformation of θ_t, $\phi_t = A^{-1}\theta_t$, for $t = 0, \ldots, n$, where the working parameter, A, is an element of $\mathcal{A}$, the class of $(p \times p)$ invertible matrices. Thus we generalize (16) to

$$\begin{aligned} \text{Observation equation: } & y_t = X_t'\beta + F_t'A\phi_t + \nu_t, \\ & \nu_t \sim N(0, \sigma^2); \\ \text{System equation: } & \phi_t = A^{-1}GA\,\phi_{t-1} + \tilde{\omega}_t, \\ & \tilde{\omega}_t \sim N(0, \tilde{T}); \\ \text{Initial condition: } & \phi_0 \sim N(0, \kappa\tilde{T}), \end{aligned} \tag{18}$$

where $\tilde{T} = A^{-1}T(A^{-1})'$. Model (18) reduces to model (16) when $A = A_0 \equiv I_p$ and it is easy to verify that $p(Y_{\text{obs}} \mid \xi)$ is identical for model (16) and model (18) for all $A \in \mathcal{A}$, i.e., condition (3) holds. Thus, applying PXEM to this formulation of the model will result in an algorithm with rate of convergence at least as good as that of Shumway and Stoffer's algorithm.

4.4. *The PXEM algorithm for maximum likelihood*

To implement the PXEM algorithm for ML calculations using (18), we define $Y_{\text{aug}} = \{\phi_0, (y_t, \phi_t), t = 1, \ldots, n\}$ and thus

$$Q_{\text{PXEM}}(\xi, A \mid \xi^{(l)}, A_0) \tag{19a}$$
$$= -\frac{1}{2\sigma^2}\sum_{t=1}^{n}\{(y_t - X_t'\beta - F_t'Ah_t)^2 + F_t'AH_t(F_t'A)'\}$$
$$-\frac{n}{2}\log\sigma^2 - \frac{n+1}{2}\log|\tilde{T}| \tag{19b}$$
$$-\frac{1}{2}\text{tr}[\tilde{T}^{-1}\{(H_0 + h_0h_0')/\kappa + D - C\tilde{G}' - \tilde{G}C' + \tilde{G}B\tilde{G}'\}], \tag{19c}$$

where $\tilde{G} = A^{-1}GA$. The E-step involves computing the various quantities in (19) and is computationally equivalent to E-step of EM. Because of the dependence of $\tilde{G}$ on A, the M-step, which maximizes $Q_{\text{PXEM}}(\xi, A \mid \xi^{(l)}, A_0)$ jointly as a function of ξ and A is a non-trivial task. However, we can typically impose structure on A such that $\tilde{G} = G$, i.e., if we choose A such that $GA = AG$, the trace in (19c) no longer depends on A. In this case, A (along with β) takes the role of a fixed-effect or regression coefficient in (19) and the M-step is straightforward. Iteration $(l+1)$ of PXEM has the form

E-step: Compute $\{h_t(\xi^{(l)}), H_t(\xi^{(l)}), P_t(\xi^{(l)}), t = 1, \ldots, n\}$, $B(\xi^{(l)}), C(\xi^{(l)})$ and $D(\xi^{(l)})$ using the forward-backward algorithm as described in Section 4.2;

M-step: Set

$$\begin{aligned}&\big(\beta^{(l+1)}, A^{(l+1)}\big)\\ &= \underset{(\beta,A)}{\arg\max} \sum_{t=1}^{n} \big\{\big(y_t - X_t'\beta^{(l)} - F_t'Ah_t\big)^2 + F_t'AH_t(F_t'A)'\big\},\end{aligned} \tag{20}$$

$$\begin{aligned}(\sigma^2)^{(l+1)} &= \frac{1}{n}\sum_{t=1}^{n}\big\{\big(y_t - F_t'A^{(l+1)}h_t\big)^2\\ &\quad + F_t'A^{(l+1)}H_t\big(F_t'A^{(l+1)}\big)'\big\},\end{aligned}$$

and

$$T^{(l+1)} = A^{(l+1)}\tilde{T}^{(l+1)}\big(A^{(l+1)}\big)', \tag{21}$$

where $\tilde{T}^{(l+1)}$ is given by the right-hand-side of (17).

The constraint, $GA = AG$ is always satisfied if $A = \alpha I$, where α is a scalar working parameter. In this case, (20) reduces to

$$\begin{aligned}&\begin{pmatrix}\beta^{(l+1)}\\ \alpha^{(l+1)}\end{pmatrix}\\ &= \left\{\sum_{t=1}^{n}\begin{pmatrix}X_tX_t' & X_tF_t'h_t\\ F_t'h_tX_t' & F_t'(H_t + h_th_t')F_t\end{pmatrix}\right\}^{-1}\sum_{t=1}^{n}\begin{pmatrix}X_t\\ F_t'h_t\end{pmatrix}y_t.\end{aligned} \tag{22}$$

That is, we regress y_t on X_t and $F_t'\phi_t$, accounting for the missingness of ϕ_t. Thus, this choice of working parameter ensures an easy to use algorithm which is at least as fast as the standard EM algorithm in terms of its global rate of convergence. We can often improve on this algorithm, however, by increasing the dimension of the free parameters in A. Thus, in model (18), our goal is to find the working parameter with the largest dimension that retains a simple closed form algorithm. In Appendix A, we show how A can be constrained in accordance with the form of G to ensure G and A commute and that the M-step reduces to solving the simple quadratic form in (20). Additionally, if T is subject to a constraint, we generally require A to be chosen so that $\tilde{T}$ is subject to the same constraint. This simplifies the M-step because it avoids maximization subject to the awkward constraint on A and $\tilde{T}$ that $A\tilde{T}A'$ has a particular form. For example, if we constrain T to be diagonal, requiring A to be diagonal as well avoids maximizing (19) subject to the awkward constraint that $A\tilde{T}A'$ is diagonal.

4.5. *The OSL-PXEM algorithm for Bayesian calculations*

We consider computing posterior modes using the independent semi-conjugate prior distributions $\beta \mid \sigma^2 \sim N(\mu_\beta, \sigma^2\Sigma_\beta), \sigma^2 \sim \nu\sigma_0^2/\chi_\nu^2$, and T distributed as inverse Wishart with η degrees of freedom and scale T_0. (We parameterize the inverse Wishart so that $E(T) = (\eta - p - 1)^{-1}T_0$, where p is the dimension of T.) In the OSL-PXEM algorithm, we use a one-step-late iteration to compute a proposed parameter update. In the following E-step the log posterior distribution is produced as a by-product. If the proposed update does not increase the log posterior distribution, we discard it and recompute the update using the M-step of the standard EM algorithm. Notice, that if the proposal is accepted, the E-step for the following iteration need not be recomputed.

At iteration $(l+1)$, the OSL-PXEM iteration has the following structure

E-step: If necessary, compute $\ell(\xi^{(l)} \mid y)$ using the "forward" part of the forward-backward algorithm; compute $\{h_t(\xi^{(l)}), H_t(\xi^{(l)}), P_t(\xi^{(l)}), \text{for } t = 1, \ldots, n\}, B(\xi^{(l)}), C(\xi^{(l)})$, and $D(\xi^{(l)})$ using the "backward" part of the forward-backward algorithm as described in Section 4.2;

M-step: Set $(\beta_{\text{prop}}^{(l+1)}, A_{\text{prop}}^{(l+1)})$ to the solution of

$$\begin{aligned}&-\frac{\partial}{\partial(\beta, A)}\frac{1}{2}\left[\sum_{t=1}^{n}\{(y_t - X_t'\beta - F_t'Ah_t)^2 + F_t'AH_t(F_t'A)'\}\right.\\ &\quad\left. + (\beta - \mu_\beta)'\Sigma_\beta^{-1}(\beta - \mu_\beta)\right] + \mathcal{C}_\alpha = 0,\end{aligned} \tag{23}$$

where the term

$$\mathcal{C}_\alpha = -(\sigma^2)^{(l)}\frac{\partial}{\partial(\beta, A)}\Big[(\eta + p + 1)\log|A| \tag{24a}$$

$$\quad + \frac{1}{2}\mathrm{tr}\big\{T_0(A^{-1})'\big(T^{(l)}\big)^{-1}A^{-1}\big\}\Big]\Big|_{A=A_0} \tag{24b}$$

corresponds to the prior distribution on T; set

$$\begin{aligned}&(\sigma^2)_{\text{prop}}^{(l+1)}\\ &= \frac{1}{n+q+\nu+2}\Big[\nu\sigma_0^2 + \big(\beta_{\text{prop}}^{(l+1)} - \mu_\beta\big)'\Sigma_\beta^{-1}\big(\beta_{\text{prop}}^{(l+1)} - \mu_\beta\big)\end{aligned} \tag{25a}$$

$$+ \sum_{t=1}^{n}\big\{\big(y_t - X_t'\beta_{\text{prop}}^{(l+1)} - F_t'A_{\text{prop}}^{(l+1)}h_t\big)^2 \tag{25b}$$

$$+ F_t'A_{\text{prop}}^{(l+1)}H_t\big(F_t'A_{\text{prop}}^{(l+1)}\big)'\big\}\Big], \tag{25c}$$

$$\tilde{T}_{\text{prop}}^{(l+1)} = \frac{1}{n+\eta+p+2}(T_0 + V), \tag{26}$$

and $T_{\text{prop}}^{(l+1)} = A_{\text{prop}}^{(l+1)}\tilde{T}_{\text{prop}}^{(l+1)}(A_{\text{prop}}^{(l+1)})'$;

Correction-step: Compute $\ell(\xi^{(l+1)} \mid Y)$ using the "forward" part of the forward-backward algorithm; if the proposed iterate increases the log posterior distribution, set $\xi^{(l+1)} = \xi_{\text{prop}}^{(l+1)}$; otherwise, set

$$\beta^{(l+1)} = \left(\Sigma_\beta^{-1} + \sum_{t=1}^{n}X_tX_t'\right)^{-1}\left(\Sigma_\beta^{-1}\mu_\beta + \sum_{t=1}^{n}X_ty_t\right), \tag{27}$$

$$(\sigma^2)^{(l+1)} = \frac{\nu\sigma_0^2 + \sum_{t=1}^{n}\left\{\left(y_t - X_t'\beta^{(l+1)} - F_t{}'h_t\right)^2 + F_t{}'H_tF_t\right\} + \left(\beta^{(l+1)} - \mu_\beta\right)'\Sigma_\beta^{-1}\left(\beta^{(l+1)} - \mu_\beta\right)}{n+q+\nu+2}, \tag{28}$$

and $T^{(l+1)} = \tilde{T}^{(l+1)}$.

The specific form of the solution to (23)–(24) is discussed in Appendix A for various component forms; the expression here assumes that A is chosen so that $AG = GA$. Since (24) is evaluated at $A = A_0$, computing $(\beta_{\text{prop}}^{(l+1)}, A_{\text{prop}}^{(l+1)})$ reduces to a slightly modified regression. The correction step requires no extra computation unless the proposed update is rejected, which is typically unusual as long as the prior distribution for T is weak relative to the likelihood. In particular, the correction step is not needed if a flat prior distribution is used for T regardless of the choice of prior distribution for (β, σ^2). In some of the component forms, T is constrained, e.g., to be diagonal. This affects both (24) and (26); details are given in Appendix A.

The EM algorithm for computing the MAP estimate is immediate from the computations above. In particular, iteration $(l+1)$ of EM is given by

E-step: Compute $\{h_t(\xi^{(l)}), H_t(\xi^{(l)}), P_t(\xi^{(l)}),$ for $t = 1, \ldots, n\}$, $B(\xi^{(l)})$, $C(\xi^{(l)})$, and $D(\xi^{(l)})$ using the forward-backward algorithm as described in Section 4.2;

M-step: Compute $\beta^{(l+1)}$ and $(\sigma^2)^{(l+1)}$ using (27)–(28) and set $T^{(l+1)}$ to the right-hand-side of (26).

The Shumway and Stoffer algorithm is the special case resulting from flat prior distributions. Although the EM iteration is simpler, we shall see in the next section that the working parameter speeds up the algorithm substantially while maintaining monotone convergence.

5. Computational performance

5.1. *Modeling the volume of the Nile river*

We begin by illustrating the dramatic improvement that PXEM can offer over EM when fitting a DLM via ML. Figure 3 illustrates the annual volume of the Nile river from 1871 to 1970 (Cobb 1978). The dotted lines represent the average level for two time periods: 1871–1898 and 1899–1970; the sharp drop in volume in 1899 is attributed to a dramatic change in tropical rainfall.

The computational efficiency of PXEM can be seen in an analysis of the data from 1900 to 1970. Because of the stability of the volume in this time period, we fit a first order polynomial trend DLM (see Table 1 and Appendix A.1),

$$y_t = \theta_t + \nu_t, \qquad \nu_t \sim N(0, \sigma^2), \quad \text{with} \tag{29a}$$

$$\theta_t = \theta_{t-1} + \omega_t, \qquad \omega_t \sim N(0, \tau^2), \tag{29b}$$

where y_t is river volume at time t, for $t = 1900, \ldots, 1970$. The model was fit with both EM and PXEM via ML. Both algorithms were run until the loglikelihood increased by less than 10^{-6} after starting at $(\sigma^2, \tau^2) = (12000, 55)$. PXEM converged in eight steps, while even after 10, 000 iterations of EM, the loglikelihood did not increase by as much as in four iterations of PXEM. This example illustrates the extreme improvement of PXEM that is typical of our experience in very stable systems, i.e., models with very small system variances.

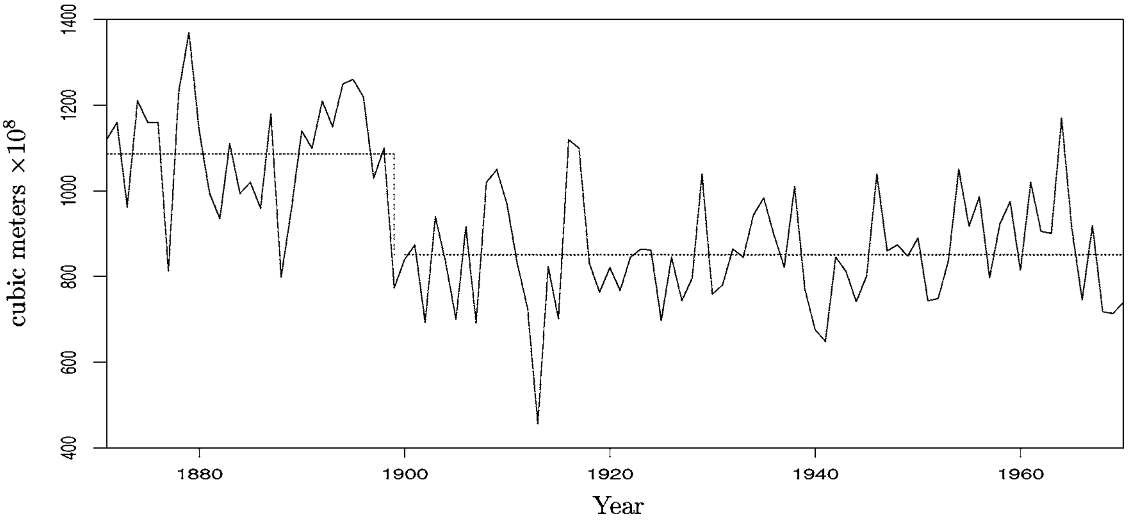

Fig. 3. *Annual volume of the Nile river: 1871–1970. The dotted lines correspond to the mean values in the time periods: 1871–1898 and 1899–1970*

In a second analysis, we added a regression component to the model, which accounts for the sharp jump in 1899 by setting x_t equal to an indicator variable for 1871–1898. We computed the MAP estimate using the entire data set and the independent prior distribution, $\sigma^2 \sim 0.01/\chi^2_{0.1}$, $\tau^2_{\rm reg} \sim 0.01/\chi^2_{0.1}$, and $p(\tau^2_{\rm pt}) \propto 1$, where the subscripts 'reg' and 'pt' indicate the system variance for the regression and polynomial trends respectively. Using the same convergence criterion as before, OSL-PXEM required 1.29 seconds (123 iterations) to converge, while EM required 3.79 seconds (469 iterations).

5.2. *Simulation studies with polynomial trend DLMs*

In this section, we describe a simulation study involving the computation of MAP estimates in a polynomial trend model. We generated 1000 data sets, each of size 20, according to a first

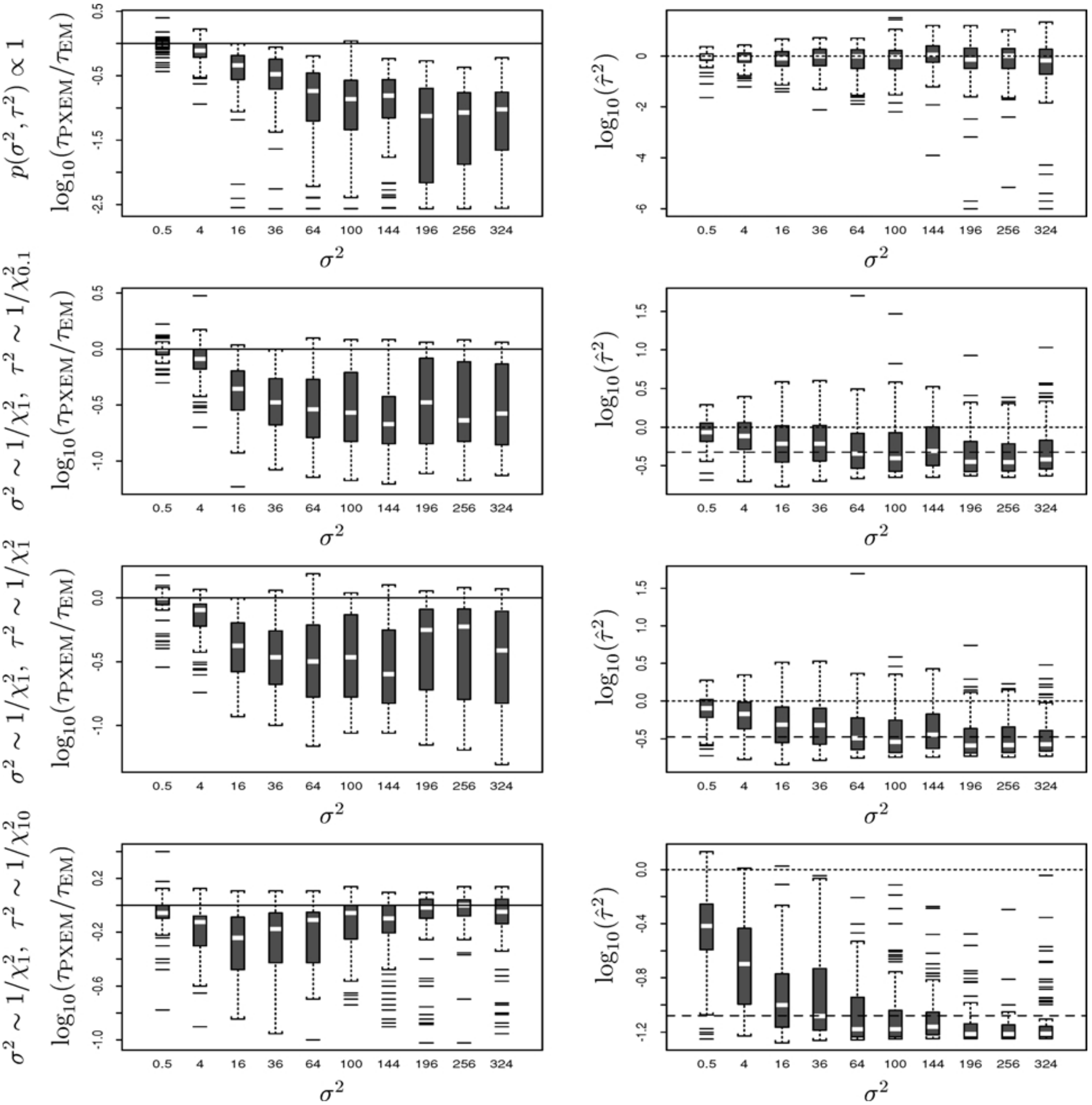

Fig. 4. *The computational gain of OSL-PXEM over EM when fitting a polynomial trend DLM. The first column plots the computational time required by OSL-PXEM relative to that required by EM; the second column illustrates the fitted value of τ^2; both are on the $\log_{10}$ scale. The rows correspond to four different prior distributions. In the second column, the dotted lines represent the true value of τ^2 and the dashed lines the prior mode. As the (true) observation variance increases the posterior mode converges to the prior mode and the computational gain of OSL-PXEM may dissipate, as in the fourth row*

Table 2. *Convergence of the loglikelihood when fitting a second order polynomial trend model with EM and PXEM. The step size of EM becomes very small long before the algorithm reaches the mode*

Iteration	EM	PXEM
start	−249.450155	−249.450155
⋮	⋮	⋮
463	−149.661955	−148.999029
464	−149.661697	−148.999029
⋮	⋮	⋮
5303	−149.264276	−148.999029
5304	−149.264276	−148.999029

order polynomial trend model (see (29)) with $\tau^2 = 1$ and 10 values of σ^2: 0.5, 4, 16, 36, 64, 100, 144, 196, 256, 324, in equal proportion. The model was fit using both EM and OSL-PXEM with four different independent prior distributions on σ^2 and τ^2: (1) $p(\sigma^2, \tau^2) \propto 1$; (2) $\sigma^2 \sim 1/\chi^2_1$, $\tau^2 \sim 1/\chi^2_{0.1}$; (3) $\sigma^2 \sim 1/\chi^2_1$, $\tau^2 \sim 1/\chi^2_1$; and (4) $\sigma^2 \sim 1/\chi^2_1$, $\tau^2 \sim 1/\chi^2_{10}$. Both EM and OSL-PXEM were started from $(\sigma^2)^{(0)} = 1$, $(\tau^2)^{(0)} = 1$. OSL-PXEM was run until the log posterior distribution increased by less than 10^{-6}; EM was stopped when the log posterior distribution reached that of the last step of PXEM. We use this convergence criterion for EM because extremely slow convergence can result in very small step sizes long before the iteration reaches the mode; an example illustrating this phenomenon when fitting a second order polynomial trend model appears in Table 2. The results appear in Fig. 4, where the rows correspond to the four prior distributions. In the first column, the boxplots show the log (base 10) of the ratio of the time required by OSL-PXEM to the time required by EM. In the second column, we plot the fitted value of τ^2 on the log (base 10) scale and indicated the true value of τ^2 and the prior mode of τ^2 with dotted and dashed lines respectively. We can see that as σ^2 grows, the prior specification becomes more influential. When the prior distribution dominates the likelihood, OSL-PXEM offers little advantage over EM, similar to the result of Section 2.3. More interestingly, when the likelihood dominates the prior, OSL-PXEM can offer significant computational gain.

Looked at another way, OSL-PXEM offers the biggest gains when the system equation is much more stable than the observation equation, i.e., when T is nearly singular. We emphasize the importance of this case. A nearly singular T indicates a very stable system equation, at least for one component. This is the ideal case for prediction. Good statistical analysis aims at including enough structure in the model to explain as much systematic variation as possible.

6. Discussion

The computational methods presented here have application well beyond the specific two-level Gaussian DLM and probit regression model. The adaptation of the one-step-late algorithm offers a general strategy for implementing PXEM with proper priors distributions, while maintaining monotone convergence. Binary time series can be modeled via a probit link that is equivalent to assuming only the sign of y_t is observed for each t. Treating y_t as missing data leads naturally to a data augmentation scheme for model fitting which can be effectively implemented using a working parameter approach in a manner analogous to that presented here. Finally, Meng and van Dyk (1999) and Liu and Wu (1999) show how the working parameter approach can improve the data augmentation algorithm for posterior sampling. Bayesian fitting of state-space and dynamic linear models would likely be improved by this methodology. Thus, the data augmentation schemes presented here promise to improve computational performance in a wide variety of state-space and dynamic linear models.

Appendix A: Details of OSL-PXEM for various forms of the DLM

In this appendix we discuss the particulars of the OSL-PXEM algorithm for the component forms given in Table 1 as well as combinations of these forms. Both the solution to (23)–(24) and the form of (26) when T is constrained depend on the component form.

We focus on computing posterior modes with the understanding that ML estimates are a special case (i.e., if we set $\Sigma_\beta = 0$, $\nu = -2$, $\sigma_0^2 = 0$, $\eta = -(p+1)$, and $T_0 = 0$). Two simplifications resulting from flat prior distributions should also be noted. If $p(T) \propto 1$, we may skip the Correction-step and if $p(\beta) \propto 1$ (i.e., $\Sigma_\beta^{-1} = 0$) we replace the denominator of (25) with $(n + \nu + 2)$. As discussed at the end of Section 4.5, we can also easily derive EM algorithms from the OSL-PXEM algorithm for both MAP and ML calculations. In this case, we use the same replacement in (25) if $\Sigma_\beta^{-1} = 0$.

A.1. *Polynomial trend models*

To fix ideas in the polynomial trend model, we begin with the linear trend model, where

$$y_t \mid \mu_t \sim N(\mu_t, \sigma^2)$$

with

$$\mu_t = \mu_{t-1} + \gamma_{t-1} + \omega_{t1}, \tag{30a}$$

$$\gamma_t = \gamma_{t-1} + \omega_{t2}, \quad (\omega_{t1}, \omega_{t2})' \sim N(0, T), \tag{30b}$$

and $\theta_t = (\mu_t, \gamma_t)'$ for $t = 1, \ldots, n$. In the absence of system variance, γ is the per unit time change in the observation mean, i.e., the slope of the linear trend. Thus, this model is used to incorporate a linear drift in the system.

It can be shown that A must be of the form

$$A_{\text{pt}} = \begin{pmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_p \\ 0 & \alpha_1 & \ldots & \alpha_{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \alpha_1 \end{pmatrix}, \tag{31}$$

to ensure A_{pt} and G commute; here the subscript 'pt' stands for polynomial trend. To derive the OSL-PXEM algorithm, we need only construct an explicit form of (23)–(24). Using (31),

$$\left.\frac{\partial \log |A_{\text{pt}}|}{\partial \alpha}\right|_{A_{\text{pt}}=A_0} = (p, 0, \ldots, 0)'$$

and

$$\left.\frac{\partial \operatorname{tr}\left\{T_0\left(A_{\text{pt}}^{-1}\right)'\left(T^{(l)}\right)^{-1}A_{\text{pt}}^{-1}\right\}}{\partial \alpha_i}\right|_{A_{\text{pt}}=A_0} = -2\operatorname{tr}\left\{\left(T^{(l)}\right)^{-1}T_0\frac{\partial A_{\text{pt}}}{\partial \alpha_i}\right\}$$

so that the correction term in (24) is

$$\mathcal{C}_\alpha = -(\sigma^2)^{(l)}\left[(\eta+p+1)\begin{pmatrix} p \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right. \tag{32a}$$

$$\left. - \begin{pmatrix} \operatorname{tr}\left\{\left(T^{(l)}\right)^{-1}T_0\partial A_{\text{pt}}/\partial\alpha_1\right\} \\ \operatorname{tr}\left\{\left(T^{(l)}\right)^{-1}T_0\partial A_{\text{pt}}/\partial\alpha_2\right\} \\ \vdots \\ \operatorname{tr}\left\{\left(T^{(l)}\right)^{-1}T_0\partial A_{\text{pt}}/\partial\alpha_p\right\} \end{pmatrix}\right], \tag{32b}$$

where $\alpha = (F_t' A_{\text{pt}})' = (\alpha_1, \alpha_2, \ldots, \alpha_p)'$. Thus, the solution of (23)–(24) is given by

$$\begin{pmatrix} \beta_{\text{prop}}^{(l+1)} \\ \alpha_{\text{prop}}^{(l+1)} \end{pmatrix} = \left\{u\left(\Sigma_\beta^{-1}\right) + \sum_{t=1}^{n}\begin{pmatrix} X_tX_t' & X_th_t' \\ h_tX_t' & H_t + h_th_t' \end{pmatrix}\right\}^{-1} \tag{33a}$$

$$\times\left\{\begin{pmatrix} \Sigma_\beta^{-1}\mu_\beta \\ \mathcal{C}_\alpha \end{pmatrix} + \sum_{t=1}^{n}\begin{pmatrix} X_t \\ h_t \end{pmatrix} y_t\right\}, \tag{33b}$$

where $u(\Sigma_\beta^{-1}) = \begin{pmatrix} \Sigma_\beta^{-1} & 0 \\ 0 & 0 \end{pmatrix}$, and $A_{\text{prop}}^{(l+1)}$ can be obtained accordingly using (31). The significant computational advantage of the OSL-PXEM algorithm for the polynomial trend model is illustrated in Sections 5.1 and 5.2.

A.2. *Polynomial growth model*

In this section we briefly describe a popular variant of the polynomial trend model known as the polynomial growth model which, except for a constraint on T, is identical to the trend model; this form of the polynomial model has long been popular in the literature (e.g., Harrison 1965, 1967). For clarity, we again consider the second order model and rewrite (30) as

$$\mu_t = \mu_{t-1} + \gamma_t + \tilde{\omega}_{t1} \quad \text{and} \quad \gamma_t = \gamma_{t-1} + \tilde{\omega}_{t2},$$

and model $(\tilde{\omega}_{t1}, \tilde{\omega}_{t2})'$ as independent Gaussian random variables (see West and Harrison (1997), Chapter 7). For the model of order p, the assumption of independence on $\tilde{\omega}$ corresponds to writing $T = U_p\Delta U_p'$ in (30) or (16), where U_p is a $(p \times p)$ upper triangular matrix with non-zero elements equal to one, and Δ is a diagonal parameter matrix.

Because of the constraint on T, we replace its usual prior with $\delta_i \overset{\text{indep}}{\sim} \nu\delta_{0i}/\chi_\nu^2$, where the diagonal of Δ is $\{\delta_1, \ldots, \delta_p\}$, i.e., $p(T) \propto |U^{-1}T(U^{-1})'|^{-(\nu/2+1)}\exp\{-\frac{1}{2}\operatorname{tr}(\Delta_0U'T^{-1}U)\}$, where Δ_0 is a diagonal matrix with elements $\{\delta_{01}, \ldots, \delta_{0p}\}$. To ensure $\tilde{T}$ satisfies the same constraint as T, we replace A_{pt} with $A_{\text{pg}} = \alpha I$, where α is a scalar working parameter. These two replacements affect the computation of $\beta_{\text{prop}}^{(l+1)}$, $A_{\text{prop}}^{(l+1)}$, and $T_{\text{prop}}^{(l+1)}$ in the OSL-PXEM algorithm. In particular, (23)–(24) is satisfied by

$$\begin{pmatrix} \beta_{\text{prop}}^{(l+1)} \\ \alpha_{\text{prop}}^{(l+1)} \end{pmatrix} = \left\{u\left(\Sigma_\beta^{-1}\right) + \sum_{t=1}^{n}\begin{pmatrix} X_tX_t' & X_th_{t1}' \\ h_{t1}X_t' & (H_t)_{11} + h_{t1}h_{t1}' \end{pmatrix}\right\}^{-1} \tag{34a}$$

$$\times\left\{\begin{pmatrix} \Sigma_\beta^{-1}\mu_\beta \\ \mathcal{C}_\alpha \end{pmatrix} + \sum_{t=1}^{n}\begin{pmatrix} X_t \\ h_{t1} \end{pmatrix} y_t\right\}, \tag{34b}$$

where h_{t1} is the first component of h_t, $(H_t)_{11}$ is the $(1, 1)$ element of H_t, and

$$\mathcal{C}_\alpha = -(\sigma^2)^{(l)}\left[(\nu+2)p - \operatorname{tr}\left\{\Delta_0U'\left(T^{(l)}\right)^{-1}U\right\}\right],$$

and (26) is replaced by

$$\tilde{T}^{(l+1)} = \frac{1}{n+\nu+3}U_p \operatorname{diag}\left\{\Delta_0 + U_p^{-1}V\left(U_p^{-1}\right)'\right\}U_p', \tag{35}$$

where $\operatorname{diag}(M)$ is a diagonal matrix with diagonal elements equal to those of the matrix M.

The difficulty with using the larger working parameter A_{pt} for the polynomial growth model, is that it is difficult to maximize $Q_{\text{PXEM}}(\xi, A \mid \xi^{(l)}, A_0)$ under the constraint on T. This maximization can be approximated using the standard parametric transformation, $(\beta, \sigma^2, \tilde{T}, A)$, by first computing $A_{\text{prop}}^{(l+1)}$ by maximizing (19) over A without constraint and then over $\tilde{T}$ subject to the constraint that $U_p^{-1}A_{\text{prop}}^{(l+1)}\tilde{T}(A_{\text{prop}}^{(l+1)})'(U_p^{-1})'$ be diagonal. This involves computing $(\beta_{\text{prop}}^{(l+1)}, A_{\text{prop}}^{(l+1)})$ using (33) and $\tilde{T}_{\text{prop}}^{(l+1)}$ using (35) with U_p replaced by $(A_{\text{prop}}^{(l+1)})^{-1}U_p$. Because this is only an approximate M-step, the Correction-step is required to guarantee monotone convergence even when a flat prior is used on T.

A.3. *Seasonal effects models*

In the seasonal effects model, we include $p+1$ "seasonal" effects, e.g., 12 months or 4 quarters. Since seasonal effects are generally used in conjunction with a polynomial trend or growth model which accounts for overall trends over time, we assume the seasonal effects sum to zero, i.e., $1_{p+1}'\theta_t = 0$. The system equation, uses the permutation matrix, G, (see Table 1) to appropriately reorder the components of θ at each time period. Because of the symmetry of the system, we generally assume the system variance can be written $T = \tau^2 I$.

To account for the constraint on θ_t, we focus on the first p elements of θ_t, $\tilde{\theta}_t = (\theta_1, \ldots, \theta_p)'$ and use the equivalent model formulation $\theta_{t+1} \mid \theta_t \sim N(G_{\text{se}}\theta_t, T)$, where

$$G_{\text{se}} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ -1 & -1 & -1 & \dots & -1 \end{pmatrix},$$

and

$$T = \tau^2\{I_p - 1_p 1_p'/(p+1)\},$$

with G_{se} and T $(p \times p)$ matrices. In the observation equation, we set F_t to the $(p \times 1)$ vector $(1, 0, \dots, 0)'$. The standard prior distribution for T is replaced with $\tau^2 \sim \eta\tau_0^2/\chi^2_\eta$, see West and Harrison (1997, Section 8.4.4).

To fit the seasonal effect model with the OSL-PXEM algorithm we require $A_{\text{se}} = \alpha I$ to account for the structure in G_{se} and the constraint on T. The algorithm is easily formulated by solving (23)–(24) using (34) with

$$\mathcal{C}_\alpha = -(\sigma^2)^{(l+1)}\left\{(\eta+2) - \frac{\eta\tau_0^2}{(\tau^2)^{(l)}}\right\}, \tag{36}$$

replacing (26) with

$$(\tilde{\tau}^2)^{(l+1)}_{\text{prop}} = \frac{\eta\tau_0^2 + \text{tr}[\{I_p - 1_p 1_p'/(p+1)\}V]}{p(n+1)+\nu+2}, \tag{37}$$

and setting $(\tau^2)^{(l+1)}_{\text{prop}} = (\tilde{\tau}^2)^{(l+1)}_{\text{prop}}(\alpha^{(l+1)}_{\text{prop}})^2$.

A.4. *Harmonic models*

Harmonic analysis models use trigonometric functions to represent cyclic trends, e.g., seasonal trends. We focus on the DLM representation of a single harmonic function with period $2\pi/\omega$ given in Table 1, where ω is a fixed constant. Higher order harmonic functions can be obtained by blocking harmonic functions with different periods (e.g., Pole, West and Harrison 1994), see Appendix A.6.

It can be shown that A must be of the form

$$A_{\text{ha}} = \begin{pmatrix} \alpha_1 & \alpha_2 \\ -\alpha_2 & \alpha_1 \end{pmatrix}$$

to ensure A_{ha} and G commute; here the subscript 'ha' indicates the harmonic analysis model. Below we discuss constraints on T, but first we allow T to be an arbitrary (2×2) positive definite matrix, in which case we need only construct an explicit form of (23)–(24) for the OSL-PXEM algorithm. Specifically, we note

$$\left.\frac{\partial \log|A_{\text{ha}}|}{\partial\alpha}\right|_{A_{\text{ha}}=A_0} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

and

$$\left.\frac{\partial\text{tr}\{T_0(A_{\text{ha}}^{-1})'(T^{(l)})^{-1}A_{\text{ha}}^{-1}\}}{\partial\alpha_i}\right|_{A_{\text{ha}}=A_0} = -2\text{tr}\left\{(T^{(l)})^{-1}T_0\frac{\partial A_{\text{ha}}}{\partial\alpha_i}\right\},$$

where $\alpha = (F_t'A_{\text{ha}})' = (\alpha_1, \alpha_2)'$, so the solution of (23)–(24) is given by (33) with

$$\mathcal{C}_\alpha = -(\sigma^2)^{(l)}\left[2(\nu+3)\begin{pmatrix} 2 \\ 0 \end{pmatrix} - 2\begin{pmatrix} \text{tr}\{(T^{(l)})^{-1}T_0\partial A_{\text{ha}}/\partial\alpha_1\} \\ \text{tr}\{(T^{(l)})^{-1}T_0\partial A_{\text{ha}}/\partial\alpha_2\} \end{pmatrix}\right].$$

If we require T to be diagonal, we replace its usual prior distribution with $\tau_i^2 \overset{\text{indep}}{\sim} \nu\tau_{0i}^2/\chi^2_\nu$, for $i = 1, 2$ and replace A_{ha} with $A_{\text{ha}} = \alpha I$ to ensure $\tilde{T}$ satisfies the same constraint as T. In the resulting OSL-PXEM (23)–(24) are satisfied using (34), with $\mathcal{C}_\alpha = -2(\sigma^2)^{(l)}[\nu + 2 - \text{tr}\{T_0(T^{(l)})^{-1}\}]$, where $T_0 = \text{diag}\{\tau_{01}, \tau_{02}\}$ and (26) is replaced with (35) with $\Delta_0 = T_0$ and $U_p = I$.

A.5. *Regression models*

In regression models, in addition to the fixed effects, we correct for p observed covariates with time varying coefficients, z_{tj}, for $t = 1, \dots, n$ and $j = 1, \dots, p$. To formalize this in model (16), we set $F_t = (z_{t1}, \dots, z_{tp})'$, $G = I$, and allow T to be an arbitrary positive definite matrix. For the OSL-PXEM algorithm, we allow $A_{\text{reg}} = (\alpha_{jk})$ to be any $(p \times p)$ invertible matrix and rewrite the observation equation in (18) as

$$y_t = X_t'\beta + \sum_{j=1}^{p}\sum_{k=1}^{p} z_{tj}\phi_{tk}\alpha_{jk} + \nu_t, \quad \nu_t \sim N(0, \sigma^2),$$

where $\phi_t = (\phi_{t1}, \dots, \phi_{tp})'$. Thus, to update A_{reg}, we treat $z_{tj}\phi_{tk}$ as p^2 regression coefficients in a linear model. In particular the solution to (23)–(24) is given by

$$\begin{pmatrix} \beta^{(l+1)}_{\text{prep}} \\ \alpha^{(l+1)}_{\text{prep}} \end{pmatrix} = \left[u(\Sigma_\beta^{-1}) + \sum_{t=1}^{n}\begin{Bmatrix} X_tX_t' & X_tE(Z_t'(\phi)) \\ E(Z_t(\phi))X_t' & E(Z_t(\phi)Z_t'(\phi)) \end{Bmatrix}\right]^{-1} \tag{38a}$$

$$\times\left[\begin{pmatrix} \Sigma_\beta^{-1}\mu_\beta \\ \mathcal{C} - \alpha \end{pmatrix} + \sum_{t=1}^{n}\begin{Bmatrix} X_t \\ E(Z_t(\phi)) \end{Bmatrix} y_t\right], \tag{38b}$$

where all expectations are conditional on $(\xi^{(l)}, A_0, y)$,

$$\begin{aligned} \alpha_{\text{reg}} &= \text{vec}(A_{\text{reg}}) \\ &\equiv (\alpha_{11}, \dots, \alpha_{1p}, \alpha_{21}, \dots, \alpha_{2p}, \dots, \alpha_{p1}, \dots, \alpha_{pp})', \end{aligned}$$

and

$$\begin{aligned} Z_t(\phi) = (&z_{t1}\phi_{t1}, \dots, z_{t1}\phi_{tp}, z_{t2}\phi_{t1}, \dots, z_{t2}\phi_{tp}, \dots, \\ &z_{tp}\phi_{t1}, \dots, z_{tp}\phi_{tp})'. \end{aligned}$$

To compute (38), note that the elements of the matrix in $E\{Z_t(\phi)Z_t'(\phi)\}$ are of the form

$$\begin{aligned} &E\left(\sum_{t=1}^{n} z_{tj}\phi_{tj'}z_{tk}\phi_{tk'} \,\middle|\, \xi^{(l)}, A_0, y\right) \\ &\quad = \sum_{t=1}^{n} z_{tj}z_{tk}\{(H_t)_{j'k'} + (h_t)_{j'}(h_t)_{k'}\}, \end{aligned}$$

where $(H_t)_{j'k'}$ is the (j', k') element of H_t and $(h_t)_{j'}$ is component j' of h_t and the components of $E\{Z_t(\phi)\}$ are of the form

$$E\left(\sum_{t=1}^{n} z_{tj}\phi_{tj'}y_t \,\middle|\, \xi^{(l)}, A_0, y\right) = \sum_{i=1}^{n} z_{ij}(h_t)_{j'}y_t.$$

The calculation required by (38b) can be accomplished by the SWEEP operator and are essentially equivalent to those required for the PXEM algorithm for mixed-effect models (Meng and van Dyk 1998, van Dyk 2000b).

Finally, to compute the correction term $\mathcal{C}_\alpha$ in (38), we note

$$\left.\frac{\partial \log|A_{\text{reg}}|}{\partial A_{\text{reg}}}\right|_{A_{\text{reg}}=A_0} = I$$

and

$$\left.\frac{\partial \operatorname{tr}\left\{T_0\left(A_{\text{reg}}^{-1}\right)'\left(T^{(l)}\right)^{-1}A_{\text{reg}}^{-1}\right\}}{\partial A_{\text{reg}}}\right|_{A_{\text{reg}}=A_0} = -2\left(T^{(l)}\right)^{-1}T_0$$

so that $\mathcal{C}_\alpha = -(\sigma^2)^{(l)}\mathrm{vec}\{(\nu+p+1)I-(T^{(l)})^{-1}T_0\}$. The remainder of the OSL-PXEM iteration follows exactly as in Section 4.5.

A.6. *Blocking component forms*

In typical data analyses, we sum component forms in the observation equation, e.g., a seasonal effect around a linear trend model. Such blocked models are generally easy to accommodate with EM-type algorithms. To illustrate blocking we combine a polynomial trend model of order p_1 with a seasonal effect model with p_2+1 seasons. The general method of combining two or more components is completely analogous. We begin by blocking the various elements of the model,

$$F_t = \begin{pmatrix} F_{[1]} \\ F_{[2]} \end{pmatrix}, \quad G = \begin{pmatrix} G_{[1]} & 0 \\ 0 & G_{[2]} \end{pmatrix}, \quad T = \begin{pmatrix} T_{[1]} & 0 \\ 0 & T_{[2]} \end{pmatrix},$$

and θ is $((p_1+p_2)\times 1)$, where $F_{[1]}$ and $G_{[1]}$ are given in the first row of Table 1 with $p = p_1$, $T_{[1]}$ is an arbitrary $(p_1 \times p_1)$ positive definite matrix, $F_{[2]}$ and $G_{[2]}$ are as described in Section A.3, and $T_{[2]} = \tau^2(I_{p_2} - 1_{p_2}1'_{p_2}/(p_2+1))$. We assume the prior distribution for T factors, i.e., $p(T) = p(T_{[1]})p(\tau^2)$, block the corresponding working parameters,

$$A = \begin{pmatrix} A_{\text{pt}} & 0 \\ 0 & A_{\text{se}} \end{pmatrix},$$

and set $\alpha = (\alpha_1, \ldots, \alpha_{p_1+1})'$, where the first p_1 components correspond to the unique elements of A_{pt}, $(F'_{[1]}A_{\text{pt}})'$ and the last element to the unique element of A_{se}.

The OSL-PXEM algorithm operates as in Section 4.5 with the modification that (26) updates $T_{[1]}$ and $T_{[2]}$ separately. In particular, the solution to (23)–(24) is given by

$$\begin{pmatrix} \beta_{\text{prop}}^{(l+1)} \\ \alpha_{\text{prop}}^{(l+1)} \end{pmatrix} = \left\{ u\left(\Sigma_\beta^{-1}\right) + \sum_{t=1}^{n} \begin{pmatrix} X_t X_t' & X_t \tilde{h}_t' \\ \tilde{h} X_t' & \tilde{H}_t + \tilde{h}_t \tilde{h}_t' \end{pmatrix} \right\}^{-1}$$
$$\times \left\{ \begin{pmatrix} \Sigma_\beta^{-1}\mu_\beta \\ \mathcal{C}_{\alpha[1]} \\ \mathcal{C}_{\alpha[2]} \end{pmatrix} + \sum_{t=1}^{n} \begin{pmatrix} X_t \\ \tilde{h}_t \end{pmatrix} y_t \right\},$$

where $\tilde{h}_t$ is the first $((p_1+1)\times 1)$ subvector of h_t, $\tilde{H}_t$ is upper left $((p_1+1)\times(p_1+1))$ submatrix of H_t, and $\mathcal{C}_{\alpha[1]}$ and $\mathcal{C}_{\alpha[2]}$ are the correction terms given in (32) and (36) respectively. Finally $\tilde{T}_{\text{prop}}^{(l+1)}$ is computed using (26) with the corresponding elements of T_0, h, and H and with $p = p_1$; $(\tilde{\tau}^2)_{\text{prop}}^{(l+1)}$ is computed with (37) using the corresponding elements of T_0, h and H; and

$$T_{\text{prop}[1]}^{(l+1)} = A_{\text{prop,pt}}^{(l+1)} \tilde{T}_{\text{prop}[1]}^{(l+1)} \left(A_{\text{prop,pt}}^{(l+1)}\right)',$$

and

$$(\tau^2)_{\text{prop}}^{(l+1)} = (\tilde{\tau}^2)_{\text{prop}}^{(l+1)} \left(\alpha_{\text{prop,se}}^{(l+1)}\right)^2,$$

where $A_{\text{prop,pt}}^{(l+1)}$ and $\alpha_{\text{prop,se}}^{(l+1)}$ are compiled from $\alpha_{\text{prop}}^{(l+1)}$.

Acknowledgments

This research was partially funded by NSF grants DMS-97-05157 and DMS-01-04129 and by the U.S. Census. We thank J. Cavanaugh for bringing our attention to the possibility of using PXEM to fit the dynamic linear model. We also thank the editors and referees for their many helpful comments.

References

Albert J.H. and Chib S. 1993. Bayesian analysis of binary and polychotomous response data. Journal of the American Statistical Association 88: 669–679.

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, Series B, Methodological 39: 1–37.

Foulley J.-L. and van Dyk D.A. 2000. The PX-EM algorithm for fast stable fitting of Henderson's mixed model. Genetics Selective Evolution 32: 143–163.

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

Goodrich R. and Caines P. 1979. Linear system identification from nonstationary cross-sectional data. IEEE Trans. Aut. Cont. AC-24: 403–411.

Green P.J. 1990. On use of the EM algorithm for penalized likelihood estimation. Journal of the Royal Statistical Society, Series B, Methodological 52: 443–452.

Gupta N. and Mehra R. 1974. Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. IEEE Trans. Aut. Cont. AC-19: 774–783.

Haas M. 1994. Igg subclass deposits in glomeruli of lupus and nonlupus membranous nephropathies. American Journal of Kidney Disease 23: 358–364.

Haas M. 1998. Value of igg subclasses and ultrastructural markers in predicting latent membranous lupus nephritis. Modern Pathology 11: 147A.

Harrison P. 1967. Exponential smoothing and short-term forecasting. Man. Sci. 13: 821–842.

Harrison P.J. 1965. Short-term sales forecasting. Applied Statistics 14: 102–139.

Harvey A. 1989. Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge University Press, New York.

Higdon D.M. 1998. Auxiliary variable methods for Markov chain Monte Carlo with applications. Journal of the American Statistical Association 93: 585–595.

Imai K. and van Dyk D.A. 2003. A Bayesian analysis of the multinomial probit model using marginal augmentation. Submitted to The Journal of Econometrics.

Jazwinski A. 1970. Stochastic Processes and Filtering Theory. Academic Press, New York.

Kalman R. 1960. A new approach to linear filtering and prediction problems. Trans. ASME J. of Basic Eng. 8: 35–45.

Kalman R. and Bucy R. 1961. New results in linear filtering and prediction theory. Trans. ASME J. of Basic Eng. 83: 95–108.

Ledolter J. 1979. A recursive approach to parameter estimation in regression and time series models. Communications in Statistics, Part A—Theory and Methods 8: 1227–1246.

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

Liu C. and Rubin D.B. 1994. The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence. Biometrika 81: 633–648.

Liu C. and Rubin D.B. 1995. ML estimation of the *t* distribution using EM and its extensions, ECM and ECME. Statistica Sinica 5: 19–39.

Liu C., Rubin D.B., and Wu Y.N. 1998. Parameter expansion for EM acceleration—The PXEM algorithm. Biometrika 75: 755–770.

Liu J.S. and Wu Y.N. 1999. Parameter expansion scheme for data augmentation. Journal of the American Statistical Association 94: 1264–1274.

McCulloch R. and Rossi P. 1994. An exact likelihood analysis of the multinomial probit model. Journal of Econometrics 64: 207–240.

Meng X.-L. and van Dyk D.A. 1997. The EM algorithm—An old folk song sung to a fast new tune (with discussion). Journal of the Royal Statistical Society, Series B, Methodological 59: 511–567.

Meng X.-L. and van Dyk D.A. 1998. Fast EM implementations for mixed-effects models. Journal of the Royal Statistical Society, Series B, Methodological 60: 559–578.

Meng X.-L. and van Dyk D.A. 1999. Seeking efficient data augmentation schemes via conditional and marginal augmentation. Biometrika 86: 301–320.

Pilla R.S. and Lindsay B.G. 1999. Alternative EM methods in high-dimensional finite mixtures. Biometrika submitted.

Pole A., West M., and Harrison J. 1994. Applied Bayesian Forecasting and Time Series Analysis. Chapman & Hall, London.

Shumway R.H. and Stoffer D.S. 1982. An approach to time series smoothing and forecastin using the EM algorithm. Journal of Time Series Analysis 3: 253–264.

van Dyk D.A. 2000a. Fast new EM-type algorithms with applications in astrophysics. Technical Report.

van Dyk D.A. 2000b. Fitting mixed-effects models using efficient EM-type algorithms. Journal of Computational and Graphical Statistics 9: 78–98.

van Dyk D.A. 2000c. Nesting EM algorithms for computational efficiency. Statistical Sinica 10: 203–225.

van Dyk D.A. and Meng X.-L. 2000. Algorithms based on data augmentation. In: Pourahmadi K. and Berk K. (Eds.), Computing Science and Statistics: Proceedings of the 31st Symposium on the Interface, 230–239. Interface Foundation of North America, Fairfax Station, VA.

van Dyk D.A. and Meng X.-L. 2001. The art of data augmentation. The Journal of Computational and Graphical Statistics 10: 1–111.

Watson W. and Engle R. 1983. Alternative algorithm for the estimation of dynamic factor, mimic and varying coefficient regression. Journal of Econometrics 23: 385–400.

West M. and Harrison J. 1997. Bayesian Forecasting and Dynamic Models (2nd edn.). Springer-Verlag, New York.

Wu C.F.J. 1983. On the convergence properties of the EM algorithms. The Annals of Statistics 11: 95–103.