Evaluating Exact VARMA Likelihood and Its Gradient when Data Are Incomplete

KRISTJAN JONASSON
University of Iceland
and
SEBASTIAN E. FERRANDO
Ryerson University

A detailed description of an algorithm for the evaluation and differentiation of the likelihood function for VARMA processes in the general case of missing values is presented. The method is based on combining the Cholesky decomposition method for complete data VARMA evaluation and the Sherman-Morrison-Woodbury formula. Potential saving for pure VAR processes is discussed and formulae for the estimation of missing values and shocks are provided. A theorem on the determinant of a low rank update is proved. Matlab implementation of the algorithm is in a companion article.

Categories and Subject Descriptors: G.3 [**Probability and Statistics**]: Time Series Analysis; G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra; G.4 [**Mathematical Software**]: Algorithm Design and Analysis; J.2 [**Physical Sciences and Engineering**]; J.4 [**Social and Behavioural Sciences**]: Economics

General Terms: Algorithms

Additional Key Words and Phrases: Exact likelihood function, missing values, incomplete data, ARMA, VARMA, vector autoregressive moving average model, determinant of low rank update, matrix derivative, matrix differentiation

ACM Reference Format:

Jonasson, K. and Ferrando, S. E. 2008. Evaluating exact VARMA likelihood and its gradient when data are incomplete. ACM Trans. Math. Softw., 35, 1, Article 5 (July 2008) 16 pages DOI 10.1145/1377603.1377608 htt://doi.acm.org/10.1145/1377603.1377608

1. INTRODUCTION

A key aspect for parameter estimation of autoregressive moving average (ARMA) processes is the efficient evaluation of the likelihood function for the

Authors' address: K. Jonasson, Department of Computer Science, Faculty of Engineering, University of Iceland, Hjardarhaga 4, 107 Reykjavik, Iceland; email: jonasson@hi.is.

parameters. In practice multivariate or vector valued processes (VARMA) are important, as well as the more general case of missing values. Evaluation of the gradient of the likelihood function is also important for its maximization using traditional numerical optimization.

This paper provides detailed practical formulae for calculating the value and the gradient of a VARMA likelihood function, both for the complete data case, and when there are missing values. A companion algorithm article [Jonasson 2008] presents Matlab programs that implement these formulae, along with a demonstration on how to carry out actual parameter estimation, and a report of numerical experiments with the programs. Both the ability to deal with missing values and to evaluate gradients are beyond the capabilities of previously published programs. The technique used to treat missing values is also new.

We concentrate on the *exact* likelihood function, not the conditional likelihood (where the initial shocks are assumed to be zero), both because the latter is not easily applicable when values are missing or when the model has moving average terms, and also because the exact likelihood does in many practical applications give significantly better parameter estimates. An alternative to using classical numerical optimization to maximize the likelihood is to use the EM-algorithm, which may in some situations be the method of choice in the presence of missing values. Application of the EM-algorithm to VARMA time series is discussed in a new paper [Metaxoglou and Smith 2007].

Three approaches to calculating exact likelihood of univariate ARMA processes have been described in the literature: (A) one that we shall refer to as the *presample method* described by Siddiqui [1958] for pure MA processes, (B) the *Cholesky decomposition method*, first described by Phadke and Kedem [1978], and (C) a *state space Kalman filter method* described by Harvey and Phillips [1979]. Several authors have described improvements and generalizations of the originally proposed methods, in particular, all three approaches have been generalized to multivariate models and to univariate models with missing values, and the Kalman filter method has been extended to the missing value multivariate case. An overview of the developments is given by Penzer and Shea [1997]. Among the papers discussed there are Ljung and Box [1979] describing a computationally efficient multivariate implementation of the presample method; Jones [1980], extending the Kalman filter approach to ARMA with missing values, and Ansley and Kohn [1983], describing a Kalman filter method to evaluate VARMA likelihood when values are missing. This last method has been publicized in several text books. The Penzer and Shea paper itself deals with extending the Cholesky decomposition method to the univariate missing value case. In addition to the references in Penzer and Shea [1997], Ljung [1989] discusses estimation of missing values for univariate processes, Mauricio [2002] gives details of a complete data multivariate implementation of the Cholesky decomposition method, and Mélard et al. [2006] describe estimation of structured VARMA models with complete data, allowing unit roots.

Two Fortran programs for VARMA likelihood evaluation in the complete data case have been published: the Kalman filter method is implemented by Shea [1989], and the presample method by Mauricio [1997]. In addition, pure VAR models (with complete data) may be fitted using the Matlab package ARfit,

described and published in Neumaier and Schneider [2001] and Schneider and Neumaier [2001].

The Cholesky decomposition method has some advantages. For the complete data case it is considerably simpler and more direct than the other two approaches. Both Penzer and Shea [1997] and Mauricio [2002] compare its efficiency with the Kalman filter method and find that it is faster in the important case when there are more autoregressive terms than moving average terms (cf. Table 1 on p. 925 in Penzer and Shea's paper and Table 1 on p. 484 in Mauricio's paper). As detailed by Penzer and Shea, many authors have also pointed out that for the missing value case the filtering approach may suffer from numerical instabilities, and although remedies have been suggested, they come at some computational cost.

In this article we take the Cholesky approach. To review its history briefly, the original article of Phadke and Kedem [1978] treats VMA models, extension to ARMA models is in Ansley [1979], Brockwell and Davis [1987, Ch. 11] describe a VARMA implementation (they and some other authors refer to the method as the *innovation method*), and Penzer and Shea [1997] provide a way of handling missing values in the ARMA case, albeit not the same as our way. To our knowledge, the current article is the first one to give details of extending the Cholesky decomposition method to the missing value VARMA case, as well as being the first paper to provide derivative formulae. It could be argued that it would have been more useful to give details and an associated publicly available program for the Kalman filter method. According to Ansley and Kohn [1983] missing values do not add to the computational cost of the filtering method, but with the current method many missing values are costly. However few missing values do not cost much, so judging by the results quoted in the previous paragraph our approach wins in that case.

The article is organized as follows. Section 2 introduces the basic notation and reviews the Cholesky decomposition method for the complete data case. Section 3, the main section of the paper, describes our approach to dealing with the missing value case. Finally Section 4 describes the main ideas and techniques used to compute the derivative of the likelihood function.

2. NOTATION AND THE CHOLESKY DECOMPOSITION METHOD

2.1 Model Notation

A VARMA model describing a time series of values $\mathbf{x}_t \in \mathbb{R}^r$ for integer t is given by:

$$\mathbf{x}_t - \boldsymbol{\mu} = \sum_{j=1}^{p} \boldsymbol{A}_j(\mathbf{x}_{t-j} - \boldsymbol{\mu}) + \mathbf{y}_t \tag{1}$$

where

$$\mathbf{y}_t = \boldsymbol{\varepsilon}_t + \sum_{j=1}^{q} \boldsymbol{B}_j \boldsymbol{\varepsilon}_{t-j}, \tag{2}$$

$\boldsymbol{\mu}$ is the expected value of $\mathbf{x}_t$, the A_j's and the B_j's are $r \times r$ matrices, and the $\boldsymbol{\varepsilon}_t$'s are r-variate $N(\mathbf{0}, \Sigma)$ uncorrelated in time. Let θ denote the

$(p+q)r^2+r(r+3)/2$-dimensional vector of all the parameters (the elements of the A_j's, the B_j's, Σ and $\boldsymbol{\mu}$; Σ being symmetric). If there are no missing values, observations $\mathbf{x}_t$ for $t = 1, \ldots, n$ are given, and $\mathbf{x}$ denotes the nr-vector $(\mathbf{x}_1^{\mathrm{T}}, ..., \mathbf{x}_n^{\mathrm{T}})^{\mathrm{T}}$ of all these values. When there are missing values the observations are limited to a subvector $\mathbf{x}_{\mathrm{o}} \in \mathbb{R}^N$ of $\mathbf{x}$, and $\mathbf{x}_{\mathrm{m}} \in \mathbb{R}^M$ is a vector of the missing values, say $\mathbf{x}_{\mathrm{m}} = (x_{\mathrm{m}_1}, \ldots, x_{\mathrm{m}_M})$. If the time series is stationary then the complete data log-likelihood function is given by

$$l(\theta) = -\frac{1}{2}(nr \log 2\pi + \log\det S + (\mathbf{x} - \bar{\boldsymbol{\mu}})^{\mathrm{T}} S^{-1}(\mathbf{x} - \bar{\boldsymbol{\mu}})) \tag{3}$$

where $S = cov_\theta(\mathbf{x})$ and $\bar{\boldsymbol{\mu}} = E_\theta(\mathbf{x}) = (\boldsymbol{\mu}^{\mathrm{T}}, \ldots, \boldsymbol{\mu}^{\mathrm{T}})^{\mathrm{T}}$. The log-likelihood function for the observed data is given by

$$l_{\mathrm{o}}(\theta) = -\frac{1}{2}(N \log 2\pi + \log\det S_{\mathrm{o}} + (\mathbf{x}_{\mathrm{o}} - \bar{\boldsymbol{\mu}}_{\mathrm{o}})^T S_{\mathrm{o}}^{-1}(\mathbf{x}_{\mathrm{o}} - \bar{\boldsymbol{\mu}}_{\mathrm{o}})) \tag{4}$$

where $S_{\mathrm{o}} = \mathrm{cov}_\theta(\mathbf{x}_{\mathrm{o}})$ is obtained from S by removing rows $m_1, \ldots, m_M$ and columns $m_1, \ldots, m_M$, and $\bar{\boldsymbol{\mu}}_{\mathrm{o}} = E_\theta(\mathbf{x}_{\mathrm{o}})$ is obtained from $\bar{\boldsymbol{\mu}}$ by removing components $m_1, \ldots, m_M$ (see for example [Ljung 1989]).

We have included the mean of the series among the parameters, instead of assuming a zero-mean process as is customary in the literature. This is not important when there are no missing values: one can simply subtract the mean of the series. When there are missing values, this might however cause a bias. Say a weather station was out of function during a cold spell. Then the mean of all observed temperature values would probably overestimate the true mean, but if other nearby stations were measuring during the cold spell then maximizing the likelihood of a VARMA model with the mean as a free parameter would avoid this bias.

2.2 Likelihood Evaluation for Complete Data

We now turn attention to the evaluation of (3) and proceed in a similar vein as Mauricio [2002] and Brockwell and Davis [1987] (and as briefly suggested in Penzer and Shea [1997]). From (1),

$$\mathbf{y}_t = \mathbf{x}_t - \boldsymbol{\mu} - \sum_{j=1}^{p} A_j(\mathbf{x}_{t-j} - \boldsymbol{\mu})$$

for $t > p$. Let $\mathbf{w}_t = \mathbf{x}_t - \boldsymbol{\mu}$ for $t \le p$ and $\mathbf{w}_t = \mathbf{y}_t$ for $t > p$ and let $\mathbf{w} = (\mathbf{w}_1^{\mathrm{T}}, \ldots, \mathbf{w}_n^{\mathrm{T}})^{\mathrm{T}}$. Then $\mathbf{w} = \Lambda(\mathbf{x} - \bar{\boldsymbol{\mu}})$ where Λ is the $nr \times nr$ lower triangular block-band matrix given by

$$\Lambda = \begin{bmatrix} I & & & & & \\ & \ddots & & & & \\ & & I & & & \\ -A_p & \cdots & -A_1 & I & & \\ & \ddots & & \ddots & \ddots & \\ & & -A_p & \cdots & -A_1 & I \end{bmatrix}. \tag{5}$$

Now let $C_j = \mathrm{cov}(\mathbf{x}_t, \boldsymbol{\varepsilon}_{t-j})$, $G_j = \mathrm{cov}(\mathbf{y}_t, \mathbf{x}_{t-j})$, $W_j = \mathrm{cov}(\mathbf{y}_t, \mathbf{y}_{t-j})$ and $S_j = cov(\mathbf{x}_t, \mathbf{x}_{t-j})$, (all these are $r \times r$ matrices). Note that with this notation,

$$S = \begin{bmatrix} S_0 & S_1^{\mathrm{T}} & \cdots & S_{n-1}^{\mathrm{T}} \\ S_1 & S_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & S_1^{\mathrm{T}} \\ S_{n-1} & \cdots & S_1 & S_0 \end{bmatrix}. \tag{6}$$

Furthermore, let A_i and B_j be zero for i and j outside the ranges implied by (1). By multiplying through (1) from the right with $\boldsymbol{\varepsilon}_{t-j}^{\mathrm{T}}$ for $j = 0, \ldots, q$ and taking expectations the following recurrence formulae for $C_0, C_1, C_2, \ldots$ are obtained:

$$C_j = A_1 C_{j-1} + \cdots + A_j C_0 + B_q \Sigma, \quad \text{for } j = 0, 1, \ldots \tag{7}$$

(so $C_0 = \Sigma$). With $B_0 = I$, we have by (1) and (2):

$$G_j = B_j C_0^{\mathrm{T}} + \cdots + B_q C_{q-j}^{\mathrm{T}}, \quad \text{for } j = 0, \ldots, q, \tag{8}$$

$$W_j = B_j \Sigma B_0^{\mathrm{T}} + \cdots + B_q \Sigma B_{q-j}^{\mathrm{T}}, \quad \text{for } j = 0, \ldots, q. \tag{9}$$

For $j < 0$ or $j > q$, C_j, G_j and W_j are zero. By multiplying (1) from the right with $(\mathbf{x}_{t-j} - \boldsymbol{\mu})^{\mathrm{T}}$ for $j = 0, \ldots, p$ and taking expectations one gets the following linear system (the *vector-Yule-Walker* equations) for the $r(r+1)/2 + pr^2$ elements of $S_0, \ldots, S_p$ (note that S_0 is symmetric):

$$\begin{aligned} S_0 - A_1 S_1^{\mathrm{T}} - \cdots - A_p S_p^{\mathrm{T}} &= G_0 \\ S_1 - A_1 S_0 - A_2 S_1^{\mathrm{T}} - \cdots - A_p S_{p-1}^{\mathrm{T}} &= G_1 \\ S_2 - A_1 S_1 - A_2 S_0 - A_3 S_1^{\mathrm{T}} - \cdots - A_p S_{p-2}^{\mathrm{T}} &= G_2 \\ &\vdots \\ S_p - A_1 S_{p-1} - A_2 S_{p-2} - \cdots - A_p S_0 &= G_p \end{aligned} \tag{10}$$

By substituting S_p given by the last equation into the first equation the number of unknowns is reduced by r^2. Details of the solution of (10) are given by Jónasson and Ferrando [2006]. If $q \leq p$, the covariance matrix of $\mathbf{w}$ will be given by the $nr \times nr$ matrix:

$$\Omega = \left[\begin{array}{cccc|cccccc} S_0 & S_1^{\mathrm{T}} & \cdots & S_{p-1}^{\mathrm{T}} & & & & & & \\ S_1 & S_0 & \ddots & \vdots & G_q^{\mathrm{T}} & & & & & \\ \vdots & \ddots & \ddots & S_1^{\mathrm{T}} & \vdots & \ddots & & & & \\ S_{p-1} & \cdots & S_1 & S_0 & G_1^{\mathrm{T}} & \cdots & G_q^{\mathrm{T}} & & & \\ \hline & G_q & \cdots & G_1 & W_0 & W_1^{\mathrm{T}} & \cdots & W_q^{\mathrm{T}} & & \\ & & \ddots & \vdots & W_1 & W_0 & W_1^{\mathrm{T}} & & \ddots & \\ & & & G_q & \vdots & W_1 & W_0 & \ddots & & W_q^{\mathrm{T}} \\ & & & & W_q & & \ddots & \ddots & W_1^{\mathrm{T}} & \vdots \\ & & & & & \ddots & & W_1 & W_0 & W_1^{\mathrm{T}} \\ & & & & & & W_q & \cdots & W_1 & W_0 \end{array}\right] \tag{11}$$

If $q > p$ the depiction is slightly different, the $pr \times (n-p)r$ upper right partition of Ω will be

$$\begin{bmatrix} G_p^{\mathrm{T}} & \cdots & G_q^{\mathrm{T}} & & \\ \vdots & & & \ddots & \\ G_1^{\mathrm{T}} & G_2^{\mathrm{T}} & \cdots & \cdots & G_q^{\mathrm{T}} \end{bmatrix},$$

the lower left partition will be the transpose of this, but the upper left and lower right partitions are unaltered. Since Λ has unit diagonal, one finds that

$$l(\theta) = -\frac{1}{2}(nr \log 2\pi + \log \det \Omega + \mathbf{w}^{\mathrm{T}} \Omega^{-1} \mathbf{w}) \tag{12}$$

To evaluate (12) it is most economical to calculate the Cholesky-factorization $\Omega = LL^{\mathrm{T}}$ exploiting the block-band structure and subsequently determine $\mathbf{z} = L^{-1}\mathbf{w}$ using forward substitution. Then the log-likelihood function will be given by

$$l(\theta) = -\frac{1}{2}\left(nr \log 2\pi + 2\sum_i \log l_{ii} + \mathbf{z}^{\mathrm{T}}\mathbf{z}\right). \tag{13}$$

We remark that the exposition in [Brockwell and Davis 1987] is significantly different from ours. They talk of the *innovation* algorithm but it turns out that the actual calculations are identical to the Cholesky decomposition described here.

2.3 Operation Count for Complete Data

Let $h = \max(p, q)$ and assume that $q > 0$ (see Section 3.4 for the $q = 0$ case). Given $\mathbf{x}$ it takes $r^2 p(n-p)$ multiplications to calculate $\mathbf{w}$. Determining the C_j's for $j \le q$, G_j's and W_i's with (7), (8), and (9) costs about $r^3(\min(p,q)^2/2 + q^2)$ multiplications and solving the system (10) takes roughly $r^6 p^3/3$ multiplications. The cost of the Cholesky-factorization of Ω will be about $(rh)^3/6$ multiplications for the upper left partition and $r^3(n-h)(q^2/2 + 7/6)$ for the lower partition. Finally, the multiplication count for the forward substitution for $\mathbf{z}$ is about $r^2(h^2/2 + (p/2 + q)(n-h))$.

To take an example of the savings obtained by using (13) rather than (3) let $p = q = 3$, $r = 8$ and $n = 1000$. Then Cholesky-factorization of S will cost $8000^3/6 \approx 8.5 \cdot 10^{10}$ multiplications (and take about 7 min. on a 1600 MHz Pentium M processor) but calculation with (13), including all the steps leading to it, will take $4.0 \cdot 10^6$ multiplications (and take 0.02 s).

3. MISSING VALUE CASE

We now consider the economical evaluation of the likelihood in the presence of some missing observations. As is customary, we assume that observations "are missing at random", i.e. that whether an observation is missing does not depend on its numerical value. The following two subsections consider separately the evaluation of the two nontrivial terms in (4).

3.1 Likelihood Evaluation via the Sherman-Morrison-Woodbury Formula

Consider first the term $(\mathbf{x}_o - \bar{\boldsymbol{\mu}}_o)^T S_o^{-1}(\mathbf{x}_o - \bar{\boldsymbol{\mu}}_o)$. Let $\bar{\Omega}$, $\bar{\Lambda}$ and $\bar{S}$ be obtained from Ω, Λ and S by placing rows and columns $m_1, \ldots, m_M$ after the other rows and columns and partition them as follows (with Ω_o, Λ_o and S_o being $N \times N$, and Ω_m, Λ_m and S_m being $M \times M$):

$$\bar{\Omega} = \begin{bmatrix} \Omega_o & \Omega_{om} \\ \Omega_{mo} & \Omega_m \end{bmatrix}, \bar{\Lambda} = \begin{bmatrix} \Lambda_o & \Lambda_{om} \\ \Lambda_{mo} & \Lambda_m \end{bmatrix}, \text{ and } \bar{S} = \begin{bmatrix} S_o & S_{om} \\ S_{mo} & S_m \end{bmatrix}.$$

By the definition of $\mathbf{w}$, $\Omega = \Lambda S \Lambda^T$ and therefore

$$\Omega_o = \Lambda_o S_o \Lambda_o^T + \Lambda_o S_{om} \Lambda_{om}^T + \Lambda_{om} S_{mo} \Lambda_o^T + \Lambda_{om} S_m \Lambda_{om}^T. \tag{14}$$

Λ_o is obtained from Λ by removing rows and corresponding columns, and it is therefore an invertible lower band matrix with unit diagonal and bandwidth at most rp, and Ω_o is obtained from Ω by removing rows and corresponding columns, so it is also a band matrix and its triangular factorization will be economical. It is thus attractive to operate with these matrices rather than the full matrix S_o. Defining

$$\tilde{\Omega}_o = \Lambda_o S_o \Lambda_o^T \tag{15}$$

and $\tilde{\mathbf{w}}_o = \Lambda_o(\mathbf{x}_o - \bar{\boldsymbol{\mu}})$ we have $(\mathbf{x}_o - \bar{\boldsymbol{\mu}}_o)^T S_o^{-1}(\mathbf{x}_o - \bar{\boldsymbol{\mu}}_o) = \tilde{\mathbf{w}}_o^T \tilde{\Omega}_o^{-1} \tilde{\mathbf{w}}_o$. Also, from (14) and (15)

$$\tilde{\Omega}_o = \Omega_o - \Lambda_o S_{om} \Lambda_{om}^T - \Lambda_{om} S_{om}^T \Lambda_o^T - \Lambda_{om} S_m \Lambda_{om}^T, \tag{16}$$

(keep in mind that $\bar{S}$ is symmetric). The matrices S_{om}, Λ_{om} and $S_o \Lambda_{om}$ are $N \times M$, so if the number of missing values, M, is (considerably) smaller than the number of observations, N, then (16) represents a low rank modification of Ω_o. This invites the use of the Sherman-Morrison-Woodbury (SMW) formula [Sherman and Morrison 1950; Woodbury 1950; cf. Golub and Van Loan 1983]. To retain symmetry of the matrices that need to be factorized, (16) may be rewritten as:

$$\tilde{\Omega}_o = \Omega_o + U S_m^{-1} U^T - V S_m^{-1} V^T \tag{17}$$

where $U = \Lambda_o S_{om}$ and $V = \Lambda_o S_{om} + \Lambda_{om} S_m$. It turns out that U is generally a full matrix but V is sparse, and it will transpire that it is possible to avoid forming U.

To obtain V economically, select the observed rows and missing columns from ΛS. From (5) and (6) and proceeding as when deriving (10) the following block

representation of ΛS for the case $q > p$ is obtained:

$$\Lambda S = \left[\begin{array}{ccccccccc} S_0 & S_1^{\mathrm{T}} & \cdots & & & & & S_{n-1}^{\mathrm{T}} \\ \vdots & \ddots & & & & & & \vdots \\ S_{p-1} & \cdots & S_0 & S_1^{\mathrm{T}} & \cdots & & & S_{n-p}^{\mathrm{T}} \\ \hline G_p & \cdots & G_1 & G_0 & G_{-1} & \cdots & & G_{-n+p+1} \\ \vdots & & & & \ddots & & & \vdots \\ G_q & & & & & & & \vdots \\ & \ddots & & & & & \ddots & G_{-1} \\ & & G_q & \cdots & & & & G_0 \end{array}\right].$$

For $q \le p$ the upper partition is the same, but the lower partition is:

$$\left[\begin{array}{ccccccc} G_q & \cdots & G_0 & \cdots & & G_{-n+p+1} \\ & \ddots & & \ddots & & \vdots \\ & & \ddots & & \ddots & \vdots \\ & & & G_q & \cdots & G_0 \end{array}\right].$$

For $S_{p+1}, \ldots, S_{n-1}$, multiply (1) from the right with $(\mathbf{x}_{t-j} - \boldsymbol{\mu})^{\mathrm{T}}$ for $j = p+1, \ldots, n-1$ and take expectations (as when deriving (10)), giving

$$S_j = A_1 S_{j-1} + A_2 S_{j-2} + \cdots + A_p S_{j-p} + G_j \tag{18}$$

with $G_p = 0$ for $p > q$. The G_j for negative j may be obtained using:

$$G_{-j} = C_j^{\mathrm{T}} + B_1 C_{j+1}^{\mathrm{T}} + \cdots + B_q C_{j+q}^{\mathrm{T}}$$

where the C_i's are given by the recurrence (7).

From (10) and (18) it follows that blocks (i, j) with $i > j + q$ of ΛS are zero, giving almost 50% sparsity. If the missing values are concentrated near the beginning of the observation period then V will be sparser still. This applies, for example, when the series represent measurements that did not all start at the same time, as will often be the case in practice. To take an example, if $q = 1$, $r = 2$, $n = 6$ and $\mathbf{m} = (2, 3, 4, 5, 9)$ then the sparsity pattern of V will be:

$$\left[\begin{array}{ccccc} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \\ & & & & \times \\ & & & & \times \\ & & & & \times \end{array}\right].$$

The SMW formula applied to (17) gives

$$\tilde{\Omega}_{\mathrm{o}}^{-1} = \hat{\Omega}_{\mathrm{o}}^{-1} + \hat{\Omega}_{\mathrm{o}}^{-1} V \boldsymbol{Q}^{-1} V^{\mathrm{T}} \hat{\Omega}_{\mathrm{o}}^{-1},$$

where

$$\hat{\Omega}_{\mathrm{o}} = \Omega_{\mathrm{o}} + U S_{\mathrm{m}}^{-1} U^{\mathrm{T}} \tag{19}$$

and Q is the $M \times M$ matrix $S_\mathrm{m} - V^\mathrm{T}\hat{\Omega}_\mathrm{o}^{-1}V$. Moreover, if R is the $M \times M$ matrix $S_\mathrm{m} + U^\mathrm{T}\Omega_\mathrm{o}^{-1}U$ then (again by the SMW formula):

$$\hat{\Omega}_\mathrm{o}^{-1} = \Omega_\mathrm{o}^{-1} - \Omega_\mathrm{o}^{-1}UR^{-1}U^\mathrm{T}\Omega_\mathrm{o}^{-1}.$$

If L_R is the Cholesky factor of R and $K = L_R^{-1}U^\mathrm{T}\Omega_\mathrm{o}^{-1}V$ it follows that $Q = S_\mathrm{m} - V^\mathrm{T}\Omega_\mathrm{o}^{-1}V + K^\mathrm{T}K$. The first method that springs to mind to evaluate $V^\mathrm{T}\Omega_\mathrm{o}^{-1}V$ efficiently is to Cholesky factorize $\Omega_\mathrm{o} = LL^\mathrm{T}$, use forward substitution to obtain $\hat{V} = L^{-1}V$ and form $\hat{V}^\mathrm{T}\hat{V}$. However, with this procedure $\hat{V}$ will be full and the computation of $\hat{V}^\mathrm{T}\hat{V}$ will cost $NM(M+1)/2$ multiplications. In contrast, if an $L^\mathrm{T}L$-factorization of Ω_o is employed instead of an LL^T-factorization the sparsity of V will be carried over to $\hat{V}$ with large potential savings. This is a crucial observation because, with many missing values, multiplication with $\hat{V}$ constitutes the bulk of the computation needed for the likelihood evaluation.

Thus the proposed method is: $L^\mathrm{T}L$-factorize $\Omega_\mathrm{o} = L_\mathrm{o}^\mathrm{T}L_\mathrm{o}$, and back-substitute to get $\hat{V} = L_\mathrm{o}^{-T}V$ and $\hat{\Lambda}_\mathrm{om} = L_\mathrm{o}^{-T}\Lambda_\mathrm{om}$, making use of known sparsity for all calculations (the sparsity structure of $\hat{\Lambda}_\mathrm{om}$ will be similar to that of $\hat{V}$). With $R_V = \hat{V}^\mathrm{T}\hat{V}$, $R_\Lambda = \hat{\Lambda}_\mathrm{om}^\mathrm{T}\hat{\Lambda}_\mathrm{om}$ and $P = \hat{\Lambda}_\mathrm{om}^\mathrm{T}\hat{V}$ (again exploiting sparsity) we find that $R = S_\mathrm{m} + R_V + S_\mathrm{m}R_\Lambda S_\mathrm{m} - S_\mathrm{m}P - P^\mathrm{T}S_\mathrm{m}$ (all matrices in this identity are full $M \times M$), $K = L_R^{-1}(R_V - P)$ and $Q = S_\mathrm{m} - R_V + K^\mathrm{T}K$. Let further L_Q be the Cholesky factor of Q, $\hat{\mathbf{w}}_\mathrm{o} = L_\mathrm{o}^{-1}\tilde{\mathbf{w}}_\mathrm{o}$, $\mathbf{u} = L_R^{-1}(\hat{V}^\mathrm{T}\hat{\mathbf{w}}_\mathrm{o} - S_\mathrm{m}\hat{\Lambda}_\mathrm{om}^\mathrm{T}\hat{\mathbf{w}}_\mathrm{o})$ and $\mathbf{v} = L_Q^{-1}(\hat{V}^\mathrm{T}\hat{\mathbf{w}}_\mathrm{o} - K^\mathrm{T}\mathbf{u})$. A little calculation then gives:

$$(\mathbf{x}_\mathrm{o} - \bar{\boldsymbol{\mu}}_\mathrm{o})^\mathrm{T}S_\mathrm{o}^{-1}(\mathbf{x}_\mathrm{o} - \bar{\boldsymbol{\mu}}_\mathrm{o}) = \hat{\mathbf{w}}_\mathrm{o}^\mathrm{T}\hat{\mathbf{w}}_\mathrm{o} - \mathbf{u}^\mathrm{T}\mathbf{u} + \mathbf{v}^\mathrm{T}\mathbf{v}.$$

3.2 Determinant of a Low Rank Update

Now turn attention to the other nontrivial term in (2.4), $\log\det S_\mathrm{o}$. We make use of the following elegant looking theorem.

THEOREM 1. *If A is $m \times n$, B is $n \times m$ and I_m and I_n are the m-th and n-th order identity matrices then* $\det(I_m + AB) = \det(I_n + BA)$.

PROOF. Let C and D be $m \times m$ and $n \times n$ invertible matrices such that $CAD = \left(\begin{smallmatrix} I_k & 0 \\ 0 & 0 \end{smallmatrix}\right)$ and let $D^{-1}BC^{-1} = \left(\begin{smallmatrix} B_1 & B_2 \\ B_3 & B_4 \end{smallmatrix}\right)$ be a partitioning with B_1 a $k \times k$ matrix. Then

$$\begin{aligned}
\det(I_m + AB) &= \det\left(C^{-1}C + C^{-1}\begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}D^{-1}D\begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix}C\right) \\
&= \det(C^{-1})\det\left(I_m + \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix}\right)\det C \\
&= \det\begin{bmatrix} I_k + B_1 & B_2 \\ 0 & I_{m-k} \end{bmatrix} = \det(I_k + B_1) = \det\begin{bmatrix} I_k + B_1 & 0 \\ B_3 & I_{n-k} \end{bmatrix} \\
&= \det\left(I_n + D\begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix}CC^{-1}\begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}D^{-1}\right) \\
&= \det(I_n + BA).
\end{aligned}$$

The matrices C and D may, for example, be obtained from the singular value decomposition of A.

An immediate consequence of this theorem is that the determinant of a low rank update of an arbitrary matrix M may be evaluated efficiently using $\det(M + AB^{\mathrm{T}}) = \det M \det(I + A^{\mathrm{T}}M^{-1}B)$, in this way complementing the Sherman-Morrison-Woodbury formula. Furthermore,

$$\det(X \pm AY^{-1}A^{\mathrm{T}}) = \det X \ \det Y^{-1} \det(X \pm A^{\mathrm{T}}Y^{-1}A) \tag{20}$$

From (16), (19), (20) and the definition of L_Q we now obtain $\det \tilde{\Omega}_{\mathrm{o}} = \det(\hat{\Omega} - VS_{\mathrm{m}}^{-1}V^{\mathrm{T}}) = \det \hat{\Omega}_{\mathrm{o}} \det S_{\mathrm{m}}^{-1} \det(S_{\mathrm{m}} - V^{\mathrm{T}}\hat{\Omega}_{\mathrm{o}}^{-1}V) = \det \hat{\Omega}_{\mathrm{o}} \det S_{\mathrm{m}}^{-1} \det(L_Q)^2$. Similarly, $\det \hat{\Omega}_{\mathrm{o}} = \det(\Omega_{\mathrm{o}} + US_{\mathrm{m}}^{-1}U^{\mathrm{T}}) = \det \Omega_{\mathrm{o}} \det S_{\mathrm{m}}^{-1} \det(S_{\mathrm{m}} + U^{\mathrm{T}}\Omega_{\mathrm{o}}^{-1}U) = \det \Omega_{\mathrm{o}} \det S_{\mathrm{m}}^{-1} \det(L_R)^2$. Since $\det \Lambda_{\mathrm{o}} = 1$ it now follows from (15) and the definition of L_{o} that

$$\log \det S_{\mathrm{o}} = 2(\log \det L_{\mathrm{o}} + \log \det L_R + \log \det L_Q - \log \det S_{\mathrm{m}}).$$

3.3 Estimating Missing Values and Shocks

An obvious estimate of the vector of missing values is its expected value, $\mathbf{x}_{\mathrm{m}}^E = E(\mathbf{x}_{\mathrm{m}}|\mathbf{x}_{\mathrm{o}}, \theta)$, where θ is the maximum likelihood estimate of the parameters (this is also the maximum likelihood estimate of $\mathbf{x}_{\mathrm{m}}$). Since $S_{\mathrm{mo}} = \mathrm{cov}(\mathbf{x}_{\mathrm{m}}, \mathbf{x}_{\mathrm{o}})$ and $S_{\mathrm{o}} = \mathrm{var}(\mathbf{x}_{\mathrm{o}})$,

$$\mathbf{x}_{\mathrm{m}}^E = S_{\mathrm{mo}}S_{\mathrm{o}}^{-1}(\mathbf{x}_{\mathrm{o}} - \bar{\boldsymbol{\mu}}_{\mathrm{o}}) + \bar{\boldsymbol{\mu}}_{\mathrm{m}}.$$

(where $\bar{\boldsymbol{\mu}}_{\mathrm{m}}$ consists of missing components of $\bar{\boldsymbol{\mu}}$). Similarly, the maximum likelihood estimate of the shocks ε_t is given by $\varepsilon_t^E = E(\varepsilon_t|\mathbf{x}_{\mathrm{o}}, \theta)$. For $0 \le j \le q$, $\mathrm{cov}(\varepsilon_t, \mathbf{x}_{t+j}) = C_j^{\mathrm{T}}$ and ε_t is independent of $\mathbf{x}_{t+j}$ for other j. It follows that $\boldsymbol{\varepsilon}^E = \tilde{C}S_{\mathrm{o}}^{-1}(\mathbf{x}_{\mathrm{o}} - \bar{\boldsymbol{\mu}}_{\mathrm{o}})$ where ε^E is the column vector with $\varepsilon_1^E, \ldots, \varepsilon_n^E$ and $\tilde{C}$ is obtained by removing missing columns from the $nr \times nr$ matrix:

$$\begin{bmatrix} C_0 & C_1^{\mathrm{T}} & \cdots & C_q^{\mathrm{T}} & & \\ & C_0 & & & \ddots & \\ & & \ddots & & & C_q^{\mathrm{T}} \\ & & & \ddots & & \vdots \\ & & & & C_0 & C_1^{\mathrm{T}} \\ & & & & & C_0 \end{bmatrix}.$$

With some calculation one may verify that given the matrices and vectors defined in the previous section, the estimates of $\mathbf{x}_{\mathrm{m}}$ and ε may be calculated economically using:

$$\mathbf{x}_{\mathrm{m}}^E = S_{\mathrm{m}}\mathbf{v}_2 + \bar{\boldsymbol{\mu}}_{\mathrm{m}},$$

and

$$\varepsilon^E = \tilde{C}\Lambda_{\mathrm{o}}^{\mathrm{T}}L_{\mathrm{o}}^{-T}(\hat{\mathbf{w}}_{\mathrm{o}} + \hat{V}(\mathbf{v}_1 - \mathbf{v}_2) - \hat{\Lambda}_{\mathrm{om}}S_{\mathrm{m}}\mathbf{v}_2),$$

where $\mathbf{v}_1 = L_Q^{-\mathrm{T}}\mathbf{v}$ and $\mathbf{v}_2 = L_R^{-\mathrm{T}}(\mathbf{u} + K\mathbf{v}_1)$.

3.4 Simplification for Pure Autoregressive Models

If q is zero and there are no moving average terms considerable simplification results, and it is worthwhile to review this case. Since $\mathbf{y}_t = \boldsymbol{\varepsilon}_t$ for all t the G_j and W_j matrices will all be zero apart from G_0 and W_0, which are both equal to Σ. The upper left S-partition of Ω in (2.11) will be unchanged, the G-partition will be zero and the lower-right W-partition will be a block diagonal matrix where each block is equal to Σ. For the missing value case, Ω_o needs to be Cholesky factorized. It is obtained by removing rows and corresponding columns from Ω, so that its upper left partition is the same as in the general ARMA case, but the lower right partition is a block diagonal matrix:

$$\begin{bmatrix} \Sigma_{o1} & & & \\ & \Sigma_{o2} & & \\ & & \ddots & \\ & & & \Sigma_{o,n-p} \end{bmatrix}$$

where Σ_{oi} contains rows and columns of Σ corresponding to the observed indices at time $p+i$. To obtain L_o it is therefore sufficient to Cholesky factorize Σ_{oi} for each missing pattern that occurs, which in all realistic cases will be much cheaper than Cholesky factorizing the entire Ω_o-matrix.

3.5 Operation Count for Missing Value Likelihood

Finding the C_j's, G_j's and W_j's and S_j's will be identical to the complete data case. The Cholesky factorization of Ω_o costs at most $r^2N(q^2/2+7/6)$ multiplications (unless the upper left partition is unusually big). Forming ΛS costs about $r^3(2p+q)(n-p)$ multiplications. The cost of forming $\hat{\Lambda}_{om}$ and $\hat{V}$ using back substitution depends on the missing value pattern. In the worst case, when all the missing values are at the end of the observation period the cost is approximately $rqNM$ multiplications for each, since the bandwidth of both is $\leq rq$, but as explained in section 3.1 the missing values will often be concentrated near the beginning and the cost will be much smaller. The cost of R_V, R_Λ and P also depends on the missing value pattern. In the worst case the symmetric R_V and R_Λ cost $NM^2/2$ multiplications each and P costs NM^2, but the typical cost is again much smaller (for example, with the "miss-25" pattern of Table I the cost is 5 times smaller). Next follows a series of order M^3 operations: $S_m P$ costs M^3, R costs $3M^3/2$, K and Q cost $M^3/2$ multiplications each. Finally the Cholesky factorizations for each of L_R, L_Q and $\det S_m$ cost $M^3/6$ multiplications. The multiplication count of other calculations is negligible by comparison unless M is very small. When n and M are large compared to p, q and r the governing tasks will cost $2fNM^2 + 4M^2$ multiplications where f is the savings factor of having the missing values early on.

In the pure autoregressive case the C_j's, G_j's and W_j's come for free, but solving the vector-Yule-Walker equations costs the same as before. The cost of Cholesky factorizing Ω will usually be negligible, and much cheaper than when $q > 0$. When nothing is missing, it is the number rpN of multiplications to find $\mathbf{w}$ and the number $rpN/2$ of multiplications of the forward substitution for

z that govern the computational cost. On the negative side, there will be no savings in the governing tasks when M and n are large.

4. DERIVATIVE OF THE LIKELIHOOD FUNCTION

Several different matrix operations that need to be differentiated may be identified. Matrix products are used in the calculation of **w** and the covariance matrices C_i, G_i and W_i, Cholesky factorization gives Ω, linear equations are solved to obtain the S_i-matrices and **z**, and lastly one must differentiate log det L. In the missing value case, several more matrix products, Cholesky factorizations, linear equation solutions, and determinants occur.

Nel [1980] reviews and develops matrix differentiation methods of scalar and matrix-valued functions with respect to scalars and matrices. He discusses three basic methods, and concludes that a method that he calls the *element breakdown* method is best for general purposes, and this is the approach we take. For the change of variables described in section 4.3 we also make use of his *vector rearrangement* method.

4.1 Notation for Matrix Derivatives

If f is a differentiable function on the set of $M \times N$ matrices the $M \times N$ matrix with (i, j)-element $\partial f/\partial x_{ij}$ will be denoted by $f'(X)$ or df/dX. If **f** is a vector valued function of a matrix then $d\mathbf{f}/dX$ or $\mathbf{f}'(X)$ denotes the block matrix:

$$\begin{bmatrix} \partial \mathbf{f}/\partial x_{11} & \cdots & \partial \mathbf{f}/\partial x_{1N} \\ \vdots & & \vdots \\ \partial \mathbf{f}/\partial x_{M1} & \cdots & \partial \mathbf{f}/\partial x_{MN} \end{bmatrix},$$

where each block is an m-dimensional column vector. If F is matrix valued, then dF/dX or $F'(X)$ denotes the $M \times N$ block-matrix

$$\begin{bmatrix} \partial F/\partial x_{11} & \cdots & \partial F/\partial x_{1N} \\ \vdots & & \vdots \\ \partial F/\partial x_{M1} & \cdots & \partial F/\partial x_{MN} \end{bmatrix}.$$

The (l, c)-block is an $m \times n$ matrix with (i, j)-element equal to $\partial f_{ij}(X)/\partial x_{lc}$, and will be denoted by F'_{lc} or $\left[dF/dX\right]_{lc}$. Details of matrix differentiation may be found in a technical report by Jonasson and Ferrando [2006].

4.2 Derivatives of the r × r Covariance Matrices

The matrices C_i, G_i and W_i are all simple matrix-polynomials in the parameter matrices (the A_i's, B_i's and Σ), and it is not difficult to verify that they can all be obtained by applying a sequence of operations of the following types:

$$\begin{aligned} F &\leftarrow F + XY \\ F &\leftarrow F + XY^{\mathrm{T}} \\ F &\leftarrow F + XG \\ F &\leftarrow F + XG^{\mathrm{T}} \end{aligned} \tag{21}$$

where F is the polynomial, X and Y are independent variables (parameter matrices), and G is also a polynomial obtained through such steps. Initialization can be either $F \leftarrow O$ (the $r \times r$ zero matrix) or $F \leftarrow X$ (one of the parameter matrices). Differentiation of the operations (21) are detailed in the following table, where X, Y and Z are different parameter matrices:

	Corresponding change to:		
Change to F:	$[dF/dZ]_{lc}$	$[dF/dX]_{lc}$	$[dF/dY]_{lc}$
$+XY$	0	$+\mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} Y$	$+X \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}}$
$+XY^{\mathrm{T}}$	0	$+\mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} Y^{\mathrm{T}}$	$+X \mathbf{e}_c \mathbf{e}_l^{\mathrm{T}}$
$+XG$	$+X\,[dG/dZ]_{lc}$	$+X\,[dG/dX]_{lc} + \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} G$	
$+XG^{\mathrm{T}}$	$+X\,[dG/dZ]_{lc}^{\mathrm{T}}$	$+X^{\mathrm{T}}[dG/dX]_{lc}^{\mathrm{T}} + \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} G^{\mathrm{T}}$	

For the first few applications of (21) the derivatives will be sparse, and for small p, q and/or n it may be worthwhile to exploit this sparsity. There are 5 possible sparsity patterns for dF/dX:

1) all elements are zero
2) in the (i, j)-block only the (i, j)-element is nonzero
3) only the i-th row in the (i, j)-block is nonzero
4) only the $j-$th column in the (i, j)-block is nonzero
5) the matrix is full

As an example, let $p = 1$, $q = 2$ and consider the differentiation of C_0, C_1, and C_2. These matrices are given by $C_0 = \Sigma$, $C_1 = A_1\Sigma + B_1\Sigma$ (the first operation of (21) twice) and C_2=$A_1C_1 + B_2\Sigma$ (the third operation of (21) followed by the first operation). Treating Σ as nonsymmetric to begin with, one obtains:

$$
\begin{array}{lll}
dC_0/dA_1 = 0 & [dC_1/dA_1]_{lc} = \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} \Sigma & [dC_2/dA_1]_{lc} = A_1[dC_1/dA_1]_{lc} + \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} C_2 \\
dC_0/dB_1 = 0 & [dC_1/dB_1]_{lc} = \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} \Sigma & [dC_2/dB_1]_{lc} = A_1[dC_1/dB_1]_{lc} + \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} C_2 \\
dC_0/dB_2 = 0 & dC_1/dB_2 = 0 & [dC_2/dB_2]_{lc} = \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} \Sigma \\
[dC_0/d\,\Sigma]_{lc} = \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} & [dC_1/d\,\Sigma]_{lc} = (A_1 + B_1)\mathbf{e}_l \mathbf{e}_c^{\mathrm{T}} & [dC_2/d\,\Sigma]_{lc} = A_1[dC_1/d\,\Sigma]_{lc} + B_2 \mathbf{e}_l \mathbf{e}_c^{\mathrm{T}}
\end{array}
$$

Here all the sparsity patterns are represented and the only full matrices are the derivatives of C_2 with respect to A_1 and B_1. Finally, derivatives with respect to Σ are adjusted by taking their symmetry into account.

Now we turn attention to the vector-Yule-Walker equations (10). Differentiating through these with respect to a parameter gives:

$$
\begin{aligned}
& S'_{j,lc} - (A_1 S'_{j-1,lc} + \cdots + A_j S'_{0,lc}) - (A_{j+1}(S'_{1,lc})^{\mathrm{T}} + \cdots + A_p (S'_{p-j,lc})^{\mathrm{T}}) \\
& \quad = G'_{lc} + (A'_{1,lc} S_{j-1} + \cdots + A'_{j,lc} S_0) \\
& \qquad + \left(A'_{j+1,lc} S_1^{\mathrm{T}} + \ldots + A'_p S_{p-j}^{\mathrm{T}}\right) \text{for} j = 0, \ldots, p. \qquad (22)
\end{aligned}
$$

This set of equations has exactly the same coefficient matrix as the original equations (10), but a different right hand side. It can therefore be solved to obtain the derivatives of $S_0, \ldots, S_p$ using the same factorization as was used to obtain the S_j.

4.3 Remaining Steps in Likelihood Gradient Calculation

It follows from (21) that the derivative of $\mathbf{y}_t$ (and thereby $\mathbf{w}_t$) with respect the B_j's and Σ is zero, and rules for differentiating matrix products give its derivative with respect to the A_j's and $\boldsymbol{\mu}$. For complete data, the next needed derivative is that of L, the Cholesky factor of Ω.

From $\Omega = LL^{\mathrm{T}}$ it follows that $\Omega'_{lc} = L(L'_{lc})^{\mathrm{T}} + L'_{lc}L^{\mathrm{T}}$. If Ω'_{lc}, L and L'_{lc} are partitioned as follows for a given k

$$\Omega'_{lc} = \begin{bmatrix} \Omega'_1 & & \\ \omega'^{\mathrm{T}} & \omega'_{kk} & \\ \Omega'_2 & \mathbf{t} & \Omega'_3 \end{bmatrix}, \quad L = \begin{bmatrix} L_1 & & \\ \mathbf{u}^{\mathrm{T}} & l_{kk} & \\ L_3 & \mathbf{v} & L_2 \end{bmatrix} \quad \text{and} \quad L'_{lc} = \begin{bmatrix} L'_1 & & \\ \mathbf{u}'^{\mathrm{T}} & l'_{kk} & \\ L'_3 & \mathbf{v}' & L'_2 \end{bmatrix}$$

then $2(\mathbf{u}^{\mathrm{T}}\mathbf{u}' + l_{kk}l'_{kk}) = \omega'_{kk}$ and $L_1\mathbf{u}' + L'_1\mathbf{u} = \boldsymbol{\omega}'$ so that

$$L_1\mathbf{u}' = \boldsymbol{\omega}' - L'_1\mathbf{u}$$

and

$$l'_{kk} = (\boldsymbol{\omega}'_{kk}/2 - \mathbf{u}^{\mathrm{T}}\mathbf{u}')/l_{kk}. \tag{23}$$

These relations may be used iteratively for $k = 1, 2, \ldots$ to calculate L'_{lc} line by line, with $\mathbf{u}'$ obtained from (23) with forward substitution. Care should be taken to take advantage of the block-band structure of Ω.

From $L\mathbf{z} = \mathbf{w}$ it follows that $L\mathbf{z}'_{lc} + L'_{lc}\mathbf{z} = \mathbf{w}'_{lc}$ so the derivative of $\mathbf{z}$ is given by forward substitution. To finish the calculation of the gradient of $l(\boldsymbol{\theta})$ in (13) use $d(\mathbf{z}^{\mathrm{T}}\mathbf{z})/dx_{lc} = 2\mathbf{z}^{\mathrm{T}}\mathbf{z}'_{lc}$ followed by $d(\log l_{ii})/dX = (1/l_{ii})dl_{ii}/dX$.

In the missing value case, the operations that must be differentiated are the same: matrix products, Cholesky factorization, forward substitution, and determinants of lower triangular matrices, and there is no need to give details of all of them.

4.4 Operation Count for Gradient Calculation and Possible Savings

Inspection of the formulae in Section 4.3 for the derivatives of the most costly operations, namely matrix products, Cholesky factorization and forward substitution, shows that they all cost approximately $2n_\theta$ times more multiplications than the original operations being differentiated, where $n_\theta = r^2(p+q) + r(r+1)/2$ is the total number of model parameters excluding $\boldsymbol{\mu}$ which does not enter the costly operations. The gradient calculation will therefore usually dominate the total work needed for likelihood maximization and this is confirmed by the numerical results of the companion article [Jonasson 2008].

One way of trying to reduce this work would be to use numerical gradients in the beginning iterations, when the accuracy of the gradients is not as important as closer to the solution. Using forward differencing, $(\partial/\partial\theta_k)l(\theta) = \big(l(\theta + \delta\mathbf{e}_k) - l(\theta)\big)/\delta$, the gradient can be approximated with n_θ function calls, giving a potential saving of factor 2. However, judging by the results shown in Table II in the companion article, it seems that this technique is not so useful.

Another possibility of speeding the computations exists when estimating seasonal models, structured models, or various models with constraints on the parameters such as distributed lag models. Without entering too much into

detail, such models may often be described by writing θ as a function of a reduced set of parameters, $\theta = g(\phi)$, where $\phi \in \mathbb{R}^{n_\phi}$ has (often much) fewer components than θ. The log-likelihood for a given set of parameters ϕ is $l(g(\phi))$, and the corresponding gradient is $l'(g(\phi))J_g(\phi)$, where J_g is the $n_\theta \times n_\phi$ Jacobian of the transformation g. The parameter matrices may be sparse and it would be possible to exploit the sparsity, but big savings are also possible by multiplying with the Jacobian earlier in the computation of the gradient, instead of after evaluating $l'(\theta)$. A convenient place to make the change of variables is after the differentiation of $\mathbf{w}$ and the C_j's, G_j's and W_j's. The costly derivatives come after this, so the potential saving approaches a factor of n_θ/n_ϕ. In Jonasson [2008] this course of action has been implemented, and the likelihood routines have J_g as an optional parameter.

ACKNOWLEDGMENTS

We wish to thank Jón Kr. Arason for help with proving Theorem 1 in Section 3.2.

REFERENCES

Ansley, C. F. 1979. An algorithm for the exact likelihood of a mixed autoregressive-moving average process. *Biometrika 66*, 1, 59–65.

Ansley, C. F. and Kohn, R. 1983. Exact likelihood of vector autoregressive-moving average process with missing or aggregated data. *Biometrika 70*, 1, 275–278.

Brockwell, P. J. and Davis, R. A. 1987. *Time Series: Theory and Methods*. Springer-Verlag, Berlin, Germany.

Golub, G. H. and Van Loan, C. F. 1983. *Matrix Computations*. North Oxford Academic, Oxford, UK.

Harvey, A. C. and Phillips, G. D. A. 1979. Maximum likelihood estimation of regression models with autoregressive-moving average disturbances. *Biometrika 66*, 1, 49–58.

Jonasson, K. and Ferrando, S. E. 2006. Efficient likelihood evaluation for VARMA processes with missing values. Tech. rep. VHI-01-2006 (http://hi.is/~jonasson), Faculty of Engineering, University of Iceland.

Jonasson, K. 2008. Algorithm 878: Exact VARMA likelihood and its gradient for complete and incomplete data with Matlab. *ACM Trans. Math. Softw. 35*, 1.

Jones, R. H. 1980. Maximum likelihood fitting of ARMA models to time series with missing observations. *Technometrics 22*, 3, 389–395.

Ljung, G. M. 1989. A note on the estimation of missing values in time series. *Comm. Statist.—Simul. Comput. 18*, 2, 459–465.

Ljung, G. M. and Box, G. E. P. 1979. The likelihood function of stationary autoregressive-moving average models. *Biometrika 66*, 2, 265–270.

Luceño, A. 1994. A fast algorithm for the exact likelihood of stationary and nonstationary vector autoregressive-moving average processes. *Biometrika* 81, 3, 555–565.

Mauricio, J. A. 1997. Algorithm AS 311: The exact likelihood function of a vector autoregressive moving average model. *Appl. Statist. 46*, 1, 157–171.

Mauricio, J. A. 2002. An algorithm for the exact likelihood of a stationary vector autoregressive moving average model. *J Time Series Analy. 23*, 4, 473–486.

Mélard, G., Roy, R., and Saidi, A. 2006. Exact maximum likelihood estimation of structured or unit root multivariate time series models. *Comput. Statist. Data Analy. 50*, 2957–2986.

Metaxoglou, K. and Smith, A. 2007. Maximum likelihood estimation of VARMA models using a state space EM algorithm. *J Time Series Analy. 28*, 5, 666–685.

Nel, D. G. 1980. On matrix differentiation in statistics. *South African Statistical J. 15*, 2, 137–193.

NEUMAIER, A. AND SCHNEIDER, T. 2001. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans. Math. Softw. 27*, 1, 27–57.

PENZER, J. AND SHEA, B. L. 1997. The exact likelihood of an autoregressive-moving average model with incomplete data. *Biometrika 84*, 4, 919–928.

PHADKE, M. S. AND KEDEM, G. 1978. Computation of the exact likelihood function of multivariate moving average models. *Biometrika 65*, 3, 511–19.

SCHNEIDER, T. AND NEUMAIER, A. 2001. Algorithm 808: ARfit - A Matlab package for the estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans. Math. Softw. 27*, 1, 58–65.

SHEA, B. L. 1989. Algorithm AS 242: The exact likelihood of a vector autoregressive moving average model. *Appl. Statist. 38*, 1, 161–204.

SHERMAN, J. AND MORRISON, W. J. 1950. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Statist. 21*, 124–127.

SIDDIQUI, M. M. 1958. On the inversion of the sample covariance matrix in a stationary autoregressive process. *Ann. Math. Statist. 29*, 585–588.

WOODBURY, M. A. 1950. Inverting modified matrices. Memor. rep. 42, Statistical Research Group, Princeton University, Princeton, NJ.

Received August 2006; revised March 2007, September 2007; accepted October 2007