Simulation Smoothing for State-Space Models: A Computational Efficiency Analysis

William J. McCausland *Université de Montréal, CIREQ and CIRANO

Shirley Miller †Université de Montréal

Denis Pelletier ‡North Carolina State University

Current version: December 1, 2008

Abstract

Simulation smoothing involves drawing state variables (or innovations) in discrete time state-space models from their conditional distribution given parameters and observations. Gaussian simulation smoothing is of particular interest, not only for the direct analysis of Gaussian linear models, but also for the indirect analysis of more general models. Several methods for Gaussian simulation smoothing exist, most of which are based on the Kalman filter. Since states in Gaussian linear state-space models are Gaussian Markov random fields, it is also possible to apply the Cholesky Factor Algorithm to draw states. This algorithm takes advantage of the band diagonal structure of the Hessian matrix of the log density to make efficient draws. We show how to exploit the special structure of state-space models to draw latent states even more efficiently.

*Corresponding author. Mailing address: Département de sciences économiques, C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7, Canada. Telephone: (514) 343-7281. Fax: (514) 343-7221. e-mail: william.j.mccausland@umontreal.ca. Web site: www.cirano.qc.ca/∼mccauslw.

†Same mailing address as McCausland. e-mail: shirley.miller.lira@umontreal.ca.

‡Mailing address: Department of Economics, Campus Box 8110, North Carolina State University, Raleigh, 27695-8110, USA. e-mail: denis_pelletier@ncsu.edu. Web site: http://www4.ncsu.edu/∼dpellet.

We analyse the computational efficiency of Kalman filter based methods and our new method and show that for many important cases, our method is more computationally efficient. Gains are particularly large for cases where the dimension of observed variables is large or where one makes repeated draws of states for the same parameter values. We apply our method to a multivariate Poisson model with time-varying intensities, which we use to analyse financial market transaction count data.

Key words: State space models, Markov chain Monte Carlo, Importance sampling, Count data, High frequency financial data.

1 Introduction

State space models are time series models featuring both latent and observed variables. The latent variables have different interpretations according to the application. They may be the unobserved states of a system in biology, economics or engineering. They may be time-varying parameters of a model. They may be factors in dynamic factor models, capturing covariances among a large set of observed variables in a parsimonious way.

Gaussian linear state-space models are interesting in their own right, but they are also useful devices for the analysis of more general state-space models. In some cases, the model becomes a Gaussian linear state-space model, or a close approximation, once we condition on certain variables. Such variables may be a natural part of the model, as in Carter and Kohn (1996), or they may be artificial devices, as in Kim, Shephard, and Chib (1998), Stroud, Müller, and Polson (2003) and Frühwirth-Schnatter and Wagner (2006).

In other cases, one can approximate the conditional distribution of states in a non-Gaussian or non-linear model by its counterpart in a Gaussian linear model. If the approximation is close enough, one can use the latter as an importance distribution for importance sampling, as in Durbin and Koopman (1997) or as a proposal distribution for Markov chain Monte Carlo, as in Shephard and Pitt (1997).

To fix notation, consider the following Gaussian linear state-space model, expressed using notation from de Jong and Shephard (1995):

$$y_t = X_t\beta + Z_t\alpha_t + G_t u_t, \quad t = 1, \ldots, n, \tag{1}$$

$$\alpha_{t+1} = W_t\beta + T_t\alpha_t + H_t u_t, \quad t = 1, \ldots, n-1, \tag{2}$$

$$\alpha_1 \sim \mathrm{N}(a_1, P_1), \quad u_t \sim \text{i.i.d. } \mathrm{N}(0, I_q), \tag{3}$$

where y_t is a $p \times 1$ vector of dependent variables, α_t is a $m \times 1$ vector of state variables, and β is a $k \times 1$ vector of coefficients. The matrices X_t, Z_t, G_t, W_t, T_t and H_t are known. Equation (1) is the *measurement* equation and equation (2) is the *state* equation. Let $y \equiv (y_1^\top, \ldots, y_n^\top)^\top$ and $\alpha \equiv (\alpha_1^\top, \ldots, \alpha_n^\top)^\top$.

We will consider the familiar and important question of drawing α as a block from its conditional distribution given y. There are several applications. In the case of models that are Gaussian and linear, or conditionally so, drawing states is a natural component of Gibbs sampling methods for learning about the posterior distribution of states, parameters and other functions of interest. In the case of models that are well approximated by Gaussian linear models, we can use draws for the Gaussian linear model as proposals in a Metropolis-Hastings update of states in the more general model. Shephard and Pitt (1997) show how to do this for a stochastic volatility model. We can also use draws from the Gaussian linear model for importance sampling, where the target distribution is the conditional distribution of states in the more general model. Durbin and Koopman (1997) show that this is useful for approximating the likelihood function for the more general model. Section 5 below shows an example of this.

Several authors have proposed ways of drawing states in Gaussian linear state-space models using the Kalman filter, including Carter and Kohn (1994), Frühwirth-Schnatter (1994), de Jong and Shephard (1995), and Durbin and Koopman (2002).

Rue (2001) introduces the Cholesky Factor Algorithm (CFA), an efficient way to draw Gaussian Markov Random Fields (GMRFs). He also recognizes that the conditional distribution of α given y in Gaussian linear state-space models is a special case of a GMRF. Knorr-Held and Rue (2002) comment on the relationship between the CFA and methods based on the Kalman filter.

The Kalman filter is often used to compute the likelihood function for a Gaussian linear state-space model. We can do the same using the CFA and our method. Both give evaluations of $f(\alpha|y)$ for arbitrary α with little additional computation. We can evaluate the likelihood as

$$f(y) = \frac{f(\alpha)f(y|\alpha)}{f(\alpha|y)}$$

for any value of α. A convenient choice is the conditional mean of α given y, since it is easy to obtain and simplifies the computation of $f(\alpha|y)$.

We make several contributions in this paper. In Section 2 we explicitly derive the precision (inverse of variance) and covector (precision times mean) of the conditional distribution of α given y in Gaussian linear state-space models, which are required inputs to the CFA. In Section 3, we propose a new method for drawing states in state-space models. Like the CFA, it uses the same precision and covector and does not use the Kalman filter. Unlike the CFA, it generates the conditional means $E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ and conditional variances $\mathrm{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ as a byproduct. These conditional moments turn out to be useful in an extension of the method, described in McCausland (2008), to non-Gaussian and non-linear state-space models with univariate states. In Section 4 we carefully analyze the computational efficiency of various methods for drawing states, showing that the CFA

and our new method are much more computationally efficient than methods based on the Kalman filter when p is large or when repeated draws of α are required. For the important special case of state-space models, our new method is twice as fast as CFA for large m.

We find examples of applications with large p in recent work in macroeconomics and forecasting using "data-rich" environments, where a large number of observed time series is linked to a much smaller number of latent factors. See for example Boivin and Giannoni (2006), who estimates DSGE models or Stock and Watson (1999, 2002) and Forni, Hallin, Lippi, and Reichlin (2000), who show that factors extracted from large data sets forecast better than small-scale VAR models. Examples with large numbers of repeated draws of α include the evaluation of the likelihood function in non-linear or non-Gaussian state-space models using importance sampling, as in Durbin and Koopman (1997).

Finally, we illustrate these methods using an empirical application. In Section 5, we use them to draw from the importance sampling distribution of Durbin and Koopman (1997) for approximating the likelihood function in non-linear and non-Gaussian state-space models. In our application, the measurement distribution is multivariate Poisson. Latent states govern time-varying intensities. Data are transaction counts in financial markets.

We conclude in Section 6.

2 The Precision and Covector of the Distribution $\alpha|y$

Here we derive expressions for the precision Ω and covector c of the conditional distribution of α given y, for the Gaussian linear state-space model described in equations (1), (2) and (3). The matrix Ω and vector c are required inputs for the CFA method and our new method.

Let v_t be the stacked period-t innovation:

$$v_t = \begin{bmatrix} G_t u_t \\ H_t u_t \end{bmatrix}.$$

We will assume that the variance of v_t has full rank. This is frequently, but not always, the case and we note that de Jong and Shephard (1995) and Durbin and Koopman (2002) do not require this assumption. The full rank conditional is not as restrictive as it may appear, however, for two reasons. First, we can impose linear equality restrictions on the α_t: we just simulate the α_t for the unrestricted model and use the technique of "conditioning by Kriging" to obtain draws for the restricted model. See Rue (2001) for a description in a similar context. Second, as Rue (2001) points out, state-space models where the innovation has less than full rank are usually more naturally expressed in another form, one that allows application of his CFA method. Take for example a model where a univariate latent variable α_t is an autoregressive process of order p and the measurement equation is given by (1). Such a model can be coerced into state-space form, with a p-dimensional state vector and an innovation variance of less than full rank. However, the conditional distribution of α given y is a GMRF and one can apply the CFA directly.

We define the matrix A_t as the precision of v_t and then partition it as:

$$A_t \equiv \begin{bmatrix} G_t G_t^\top & G_t H_t^\top \\ H_t G_t^\top & H_t H_t^\top \end{bmatrix}^{-1} = \begin{bmatrix} A_{11,t} & A_{12,t} \\ A_{21,t} & A_{22,t} \end{bmatrix},$$

where $A_{11,t}$ is the leading $p \times p$ submatrix. We also let $A_{22,0} \equiv P_1^{-1}$, the precision of α_1 and $A_{11,n} \equiv (G_n G_n^\top)^{-1}$, the precision of the time n innovation $G_n u_n$.

Clearly α and y are jointly Gaussian and therefore the conditional distribution of α

given y is also Gaussian. We can write the log conditional density of α given y as

$$\log f(\alpha|y) = -\frac{1}{2}\left[\alpha^\top \Omega\alpha - 2c^\top\alpha\right] + k, \tag{4}$$

where k is an unimportant term not depending on α. Using the definition of the model in equations (1), (2) and (3) we can also write

$$\log f(\alpha|y) = \log f(\alpha, y) - \log f(y) = -\frac{1}{2}g(\alpha, y) + k', \tag{5}$$

where

$$\begin{aligned} g(\alpha, y) &= (\alpha_1 - a_1)^\top P_1^{-1}(\alpha_1 - a_1) \\ &+ \sum_{t=1}^{n-1} \begin{bmatrix} y_t - X_t\beta - Z_t\alpha_t \\ \alpha_{t+1} - W_t\beta - T_t\alpha_t \end{bmatrix}^\top A_t \begin{bmatrix} y_t - X_t\beta - Z_t\alpha_t \\ \alpha_{t+1} - W_t\beta - T_t\alpha_t \end{bmatrix} \\ &+ (y_n - X_n\beta - Z_n\alpha_n)^\top (G_n G_n^\top)^{-1}(y_n - X_n\beta - Z_n\alpha_n), \end{aligned}$$

and k' is a term not depending on α.

Matching linear and quadratic terms in the α_t between equations (4) and (5), we obtain

$$\Omega \equiv \begin{bmatrix} \Omega_{11} & \Omega_{12} & 0 & \dots & 0 \\ \Omega_{21} & \Omega_{22} & \Omega_{23} & \ddots & \vdots \\ 0 & \Omega_{32} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \Omega_{n-1,n-1} & \Omega_{n-1,n} \\ 0 & \dots & 0 & \Omega_{n,n-1} & \Omega_{nn} \end{bmatrix} \quad c \equiv \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \tag{6}$$

where

$$\Omega_{tt} \equiv Z_t^\top A_{11,t} Z_t + Z_t^\top A_{12,t} T_t + T_t^\top A_{21,t} Z_t + T_t^\top A_{22,t} T_t + A_{22,t-1}, \quad t = 1, \ldots, n-1,$$

$$\Omega_{nn} \equiv Z_n^\top A_{11,n} Z_n + A_{22,n-1},$$

$$\Omega_{t+1,t} \equiv -A_{21,t} Z_t - A_{22,t} T_t, \quad t = 1, \ldots, n-1,$$

$$\Omega_{t,t+1} \equiv -Z_t^\top A_{12,t} - T_t^\top A_{22,t}, \quad t = 1, \ldots, n-1,$$

$$\begin{aligned} c_1 &\equiv (Z_1^\top A_{11,1} + T_1^\top A_{21,1})(y_1 - X_1\beta) - (Z_1^\top A_{12,1} + T_1^\top A_{22,1})(W_1\beta) \\ &+ A_{22,0}(W_0\beta + T_0\alpha_0), \end{aligned}$$

$$\begin{aligned} c_t &\equiv (Z_t^\top A_{11,t} + T_t^\top A_{21,t})(y_t - X_t\beta) - (Z_t^\top A_{12,t} + T_t^\top A_{22,t})(W_t\beta) \\ &\quad -A_{21,t-1}(y_{t-1} - X_{t-1}\beta) + A_{22,t-1}(W_{t-1}\beta), \quad t = 2, \ldots, n-1, \end{aligned}$$

$$c_n \equiv Z_n^\top A_{11,n}(y_n - X_n\beta) - A_{21,n-1}(y_{n-1} - X_{n-1}\beta) + A_{22,n-1}(W_{n-1}\beta).$$

3 Two Precision-Based Methods for Simulation Smoothing

Here we discuss two methods for state smoothing using the precision Ω and covector c of the conditional distribution of α given y. The first method is due to Rue (2001), who considers the more general problem of drawing Gaussian Markov random fields. The second method, introduced here, offers new insights and more efficient draws for the special case of Gaussian linear state-space models.

Rue (2001) introduces a simple procedure for drawing a Gaussian random vector α with a band-diagonal precision matrix Ω. We let N be the length of α and b be the number of non-zero subdiagonals. By symmetry, the bandwidth of Ω is $2b+1$. The first step is to compute the Cholesky decomposition $\Omega = LL^\top$ using an algorithm that exploits the band diagonal structure. The next step is to solve the equation $\epsilon = L^\top \alpha^*$ for α^*, where $\epsilon \sim N(0, I_N)$, using band back-substitution. Then $\alpha^* + \mu$, where μ is the mean of α, is a draw from the distribution of α. If the covector c of α is readily available but not μ, one can solve for μ in the equation $LL^\top \mu = c$ using band back-substitution twice. Rue (2001) recognizes that the vector of states α in Gaussian linear state-space models is an example of a Gaussian Markov random fields. The previous section explicitly derives Ω and c. We note that for the state-space model defined in the introduction, $N = nm$ and $b = 2m-1$.

We now introduce another method for drawing α based on the precision and covector of its conditional distribution of α given y. We draw the α_t backwards, each α_t from the distribution $\alpha_t | \alpha_{t+1}, \ldots, \alpha_n, y$. The following result, proved in Appendix A, allows us to draw α and evaluate $E[\alpha|y]$ in time n.

Result 3.1 *If $\alpha|y \sim N(\Omega^{-1}c, \Omega^{-1})$, where Ω has the block band structure of equation (6), then*

$$\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y \sim \mathrm{N}(m_t - \Sigma_t \Omega_{t,t+1}\alpha_{t+1}, \Sigma_t) \quad \textit{and} \quad E[\alpha|y] = (\mu_1^\top, \ldots, \mu_n^\top)^\top,$$

where

$$\Sigma_1 = (\Omega_{11})^{-1}, \qquad m_1 = \Sigma_1 c_1,$$

$$\Sigma_t = (\Omega_{tt} - \Omega_{t,t-1}\Sigma_{t-1}\Omega_{t-1,t})^{-1}, \qquad m_t = \Sigma_t(c_t - \Omega_{t,t-1}m_{t-1}),$$

$$\mu_n = m_n, \qquad \mu_t = m_t - \Sigma_t \Omega_{t,t+1}\mu_{t+1}.$$

The result is related to a Levinson-like algorithm introduced by Vandebril, Mastronardi, and Van Barel (2007). Their algorithm solves the equation $Bx = y$, where B is an $n \times n$ symmetric band diagonal matrix and y is a $n \times 1$ vector. Result 3.1 extends the results in Vandebril, Mastronardi, and Van Barel (2007) in two ways. First, we modify the algorithm to work with $m \times m$ submatrices of a block band diagonal matrix rather than individual elements of a band diagonal matrix. Second, we use intermediate quantities computed while solving the equation $\Omega\mu = c$ for $\mu = E[\alpha|y]$ in order to compute $E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ and $\mathrm{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$.

We can now use the following algorithm to draw α from $\alpha|y$ (MMP method hereafter).

1. Compute $\Sigma_1 = (\Omega_{11})^{-1}$, $m_1 = \Sigma_1 c_1$.

2. For $t = 2, \ldots, n$, compute $\Sigma_t = (\Omega_{tt} - \Omega_{t,t-1}\Sigma_{t-1}\Omega_{t-1,t})^{-1}$, $m_t = \Sigma_t(c_t - \Omega_{t,t-1}m_{t-1})$.

3. Draw $\alpha_n \sim \mathrm{N}(m_n, \Sigma_n)$.

4. For $t = n-1, \ldots, 1$, draw $\alpha_t \sim \mathrm{N}(m_t - \Sigma_t\Omega_{t,t+1}\alpha_{t+1}, \Sigma_t)$.

4 Efficiency Analysis

We compare the computational efficiency of various methods for drawing $\alpha|y$. We consider separately the fixed computational cost that is incurred only once, no matter how many draws are needed, and the marginal computational cost required for an additional draw. We do this because there are some applications, such as Bayesian analysis of state-space models using Gibbs sampling, in which only one draw is needed and other applications, such as importance sampling in non-Gaussian models, where many draws are needed.

We compute the cost of various matrix operations in terms of the number of floating point multiplications required per observation. All the methods listed in the introduction

have fixed costs that are third order polynomials in p and m. The methods of Rue (2001), Durbin and Koopman (2002) and the present paper all have marginal costs that are second order polynomials in p and m. We will ignore fixed cost terms of lower order than three and marginal cost terms of lower order than two. The marginal costs are only important when multiple draws are required.

We take the computational cost of multiplying an $N_1 \times N_2$ matrix by an $N_2 \times N_3$ matrix as $N_1 N_2 N_3$ scalar floating-point multiplications. If the result is symmetric or if one of the matrices is triangular, we divide by two. It is possible to multiply matrices more efficiently, but the dimensions required before realizing savings are higher than those usually encountered in state-space models. We take the cost of the Cholesky decomposition of a full $N \times N$ matrix as $N^3/6$ scalar multiplications, which is the cost using the algorithm in Press, Teukolsky, Vetterling, and Flannery (1992, p. 97). When the matrix has bandwidth $2b+1$, the cost is $Nb^2/2$. Solving a triangular system of N equations using back-substitution requires $N^2/2$ scalar multiplications. When the triangular system has bandwidth $b+1$, only Nb multiplications are required.

4.1 Fixed Costs

We first consider the cost of computing the precision Ω and covector c, which is required for the methods of Rue (2001) and the current paper.

The cost depends on how we specify the variance of v_t, the stacked innovation. The matrices G_t and H_t are more convenient for methods using the Kalman filter, while the precisions A_t are most useful for the precision-based methods. We recognize that it is often easier to specify the innovation distribution in terms of G_t and H_t rather than A_t. In most cases, however, the A_t are diagonal, constant, or take on one of a small number of values, and so the additional computation required to obtain the A_t is negligible.

There is an important case where it is more natural to provide the A_t. Linear Gaussian state-space models may be used to facilitate estimation in non-linear or non-Gaussian state-space models by providing proposal distributions for MCMC methods or importance distributions for importance sampling applications. In these cases, precisions in the approximating Gaussian model are negative Hessian matrices of the log observation density of the non-Gaussian or non-linear model. See Durbin and Koopman (1997) and Section 5 of the present paper.

In general, calculation of the Ω_{tt} and $\Omega_{t,t+1}$ is computationally demanding. However, in many cases of interest, A_t, Z_t and T_t are constant, or take on one of a small number of values. In these cases, the computational burden is a constant, not depending on n. We do need to compute each c_t, but provided that the matrix expressions in parantheses in the equations following (6) can be pre-computed, this involves matrix-vector multiplications, whose costs are only second order polynomials in p and m.

We now consider the cost of the Kalman filter, which is used in most methods for simulation smoothing. The computations are as follows:

$$e_t = y_t - [X_t\beta] - Z_t a_t, \quad D_t = Z_t P_t Z_t^\top + [G_t G_t^\top],$$

$$K_t = (T_t P_t Z_t^\top + [H_t G_t^\top]) D_t^{-1}, \quad L_t = T_t - K_t Z_t,$$

$$a_{t+1} = [W_t\beta] + T_t a_t + K_t e_t, \quad P_{t+1} = [T_t P_t] L_t^\top + [H_t H_t^\top] + [H_t G_t^\top] K_t$$

Here and elsewhere, we use braces to denote quantities that do not need to be computed for each observation. These include quantities such as $[T_t P_t]$ above that are computed in previous steps, and quantities such as $[H_t H_t^\top]$ that are usually either constant or taking values in a small pre-computable set.

Table 1 lists the matrix-matrix multiplications, Cholesky decompositions, and solutions

of triangular systems required for three high level operations: an iteration of the Kalman filter, the computation of $\Omega = LL^\top$ using standard methods for band diagonal Ω, and the computation of the Σ_t and m_t of Result 3.1. All simulation smoothing methods we are aware of use one of these high-level operations. We represent the solution of triangular systems using notation for the inverse of a triangular matrix, but no actual matrix inversions are performed, as this is inefficient. The table also gives the number of scalar multiplications for each operation as a function of p and m. Terms of less than third order are omitted, so we ignore matrix-vector multiplications, whose costs are mere second order monomials in m and p.

There are special cases where the Kalman filter computations are less costly. In some of these, the elements of T_t and Z_t are zero or one, and certain matrix multiplications do not require any scalar multiplications. In others, certain matrices are diagonal, reducing the number of multiplications by an order.

The relative efficiency of precision-based methods compared with Kalman filter based methods depends on various features of the application. We see that the precision-based methods have no third order monomials involving p. For the MMP method, the coefficient of the m^3 term is 7/6, compared with 2 for the CFA and 2 for the Kalman filter if $T_t P_t$ is a general matrix multiplication. If T_t is diagonal or composed of zeros and ones, the coefficient of m^3 drops to 1 for the Kalman filter.

4.2 Marginal Costs

Compared with the fixed cost of pre-processing, the marginal computational cost of an additional draw from $\alpha|y$ is negligible for all three methods we consider. In particular, no matrix-matrix multiplications, matrix inversions, or Cholesky decompositions are required. However, when large numbers of these additional draws are required, this marginal cost

Table 1: Scalar multiplications needed for pre-computation.

Method	Operation	Scalar multiplications
Kalman	$P_t Z_t^\top$	$m^2 p$
	$Z_t[P_t Z_t^\top]$	$mp^2/2$
	$T_t[P_t Z_t^\top]$	$m^2 p$
	$D_t = \Upsilon_t \Upsilon_t^\top$ (Cholesky)	$p^3/6$
	$[T_t P_t Z_t^\top + H_t G_t^\top](\Upsilon_t^\top)^{-1}\Upsilon_t^{-1}$	mp^2
	$K_t Z_t$	$m^2 p$
	$T_t P_t$	m^3
	$[T_t P_t] L_t^\top$	m^3
	$[H_t G_t^\top] K_t$	$m^2 p$
CFA	$\Omega = LL^\top$	$2m^3$
MMP	$(\Omega_{tt} - \Omega_{t,t-1}\Sigma_{t-1}\Omega_{t-1,t}) = \Lambda_t \Lambda_t^\top$ (Cholesky)	$m^3/6$
	$\Lambda_t^{-1}\Omega_{t,t+1}$	$m^3/2$
	$\Omega_{t+1,t}\Sigma_t\Omega_{t,t+1} = [\Lambda_t^{-1}\Omega_{t,t+1}]^\top[\Lambda_t^{-1}\Omega_{t,t+1}]$	$m^3/2$

becomes important. It is here that the precision-based methods are clearly more efficient than those based on the Kalman filter. We use the methods of Durbin and Koopman (2002) and de Jong and Shephard (1995) as benchmarks.

Using the modified simulation smoothing algorithm in Section 2.3 of Durbin and Koopman (2002) (DK hereafter), an additional draw from $\alpha|y$ requires the following computations. We define $\epsilon_t \equiv G_t u_t$ and $\eta_t \equiv H_t u_t$, and assume $G_t^\top H_t = 0$ and $X_t \beta = 0$, recognizing that these assumptions can be easily relaxed. The first step is forward simulation using equations (6) and (7) in that article.

$$x_1 \sim N(0, P_1), \quad v_t^+ = Z_t x_t + \epsilon_t^+ \quad x_{t+1} = T_t x_t - K_t v_t^+ + \eta_t^+,$$

where $\epsilon_t^+ \sim N(0, \Xi_t)$ and $\eta_t^+ \sim N(0, Q_t)$. The next step is the backwards recursion of equation (5):

$$r_n = 0, \quad r_{t-1} = [Z_t D_t^{-1}] v_t^+ + L_t^\top r_t,$$

and the computation of residuals in equation (4):

$$\hat{\eta}_t^+ = Q_t r_t.$$

A draw $\tilde{\eta}$ from the conditional distribution of η given y is given by

$$\tilde{\eta} = \hat{\eta} - \hat{\eta}^+ + \eta^+,$$

where $\hat{\eta}$ is a pre-computed vector. To construct a draw $\tilde{\alpha}$ from the conditional distribution of α given y, we use

$$\tilde{\alpha}_1 = \hat{\alpha}_1 - P_1 r_0 + x_1, \quad \tilde{\alpha}_{t+1} = T_t \tilde{\alpha}_t + \tilde{\eta}_t,$$

where $\hat{\alpha}_1$ is pre-computed.

de Jong and Shephard (1995) (DeJS hereafter) draw $\alpha|y$ using the following steps, given in equation (4) of their paper. First ϵ_t is drawn from $N(0, \sigma^2 C_t)$, where the Cholesky factor of $\sigma^2 C_t$ can be pre-computed. Then r_t is computed using the backwards recursion

$$r_{t-1} = [Z_t^\top D_t^{-1} e_t] + L_t^\top r_t - [V_t^\top C_t^{-1}]\epsilon_t.$$

Next, α_{t+1} is computed as

$$\alpha_{t+1} = [W_t \beta] + T_t \alpha_t + \Omega_t r_t + \epsilon_t.$$

In our approach, we draw, for each observation, a vector $v_t \sim N(0, I_m)$ and compute

$$\alpha_t = m_t - [\Sigma_t \Omega_{t,t+1}]\alpha_{t+1} + \Lambda_t^{-1} v_t.$$

Computing $\Lambda_t^{-1} v_t$ using Λ_t (which is triangular, see Table 1) requires $m(m-1)/2$ multiplications and m floating point divisions. If we are making multiple draws, we can compute the reciprocals of the diagonal elements of Λ_t once and convert the divisions into multiplications, which are typically much less costly.

The band back-substitution used by Rue (2001) is quite similar to this. However, it is a little less efficient if one is using standard band back-substitution algorithms. These do not take advantage of the special structure of state-space models, for which Ω has elements equal to zero in its first $2m-1$ subdiagonals.

5 An Empirical Application to Count Models

Durbin and Koopman (1997) show how to compute an arbitrarily accurate evaluation of the likelihood function for a semi-Gaussian state-space model in which the state evolves according to equation (2), but the conditional distribution of observations given states is given by a general distribution with density (or mass) function $p(y|\alpha)$. To simplify, we suppress notation for the dependence on θ, the vector of parameters.

The approach is as follows. The first step is to construct a fully Gaussian state-space model with the same state dynamics as the semi-Gaussian model but with a Gaussian measurement equation of the following form:

$$y_t = \mu_t + Z_t \alpha_t + \epsilon_t, \tag{7}$$

where the ϵ_t are independent $N(0, \Xi_t)$ and independent of the state equation innovations. The Z_t are matrices such that the distribution of y given α depends only on the $Z_t \alpha_t$. They choose μ_t and Ξ_t such that the implied conditional density $g(y|\alpha)$ approximates $p(y|\alpha)$ as a function of α near the mode of $p(\alpha|y)$. The next step is to draw a sample of size N from

the conditional distribution of α given y for the fully Gaussian state-space model. The final step is to use this sample as an importance sample to approximate the likelihood for the semi-Gaussian model.

The Gaussian measurement density $g(y|\alpha)$ is chosen such that $\log g(y|\alpha)$ is a quadratic approximation of $\log p(y|\alpha)$, as a function of α, at the mode $\hat{\alpha}$ of the density $p(\alpha|y)$. Durbin and Koopman (1997) find this density by iterating the following steps until convergence to obtain μ_t and Ξ_t:

1. Using the current values of the μ_t and Ξ_t, compute $\hat{\alpha} = E_g[\alpha|y]$, where E_g denotes expectation with respect to the Gaussian density $g(\alpha|y)$. Durbin and Koopman (1997) use routine Kalman filtering and smoothing with the fully Gaussian state-space model to find $\hat{\alpha}$.

2. Using the current $\hat{\alpha}$, compute the μ_t and Ξ_t such that $\log p(y|\alpha)$ and $\log g(y|\alpha)$ have the same gradient and Hessian (with respect to α) at $\hat{\alpha}$. Durbin and Koopman (1997) show that μ_t and Ξ_t solve the following two equations:

$$\frac{\partial \log p(y_t|\hat{\alpha}_t)}{\partial(Z_t\alpha_t)} - \Xi_t^{-1}(y_t - Z_t\hat{\alpha}_t - \mu_t) = 0, \tag{8}$$

$$\frac{\partial^2 \log p(y_t|\hat{\alpha}_t)}{\partial(Z_t\alpha_t)\partial(Z_t\alpha_t)'} + \Xi_t^{-1} = 0. \tag{9}$$

It is interesting to note that this delivers the specification of the measurement equation error of the fully Gaussian model directly in terms of the precision Ξ_t^{-1} rather than the variance Ξ_t directly.

The likelihood function $L(\theta)$ we wish to evaluate is

$$L(\theta) = p(y) = \int p(y, \alpha)d\alpha = \int p(y|\alpha)p(\alpha)d\alpha. \tag{10}$$

Durbin and Koopman (1997) employ importance sampling to efficiently and accurately approximate the above integrals. The likelihood for the approximating Gaussian model is

$$L_g(\theta) = g(y) = \frac{g(y,\alpha)}{g(\alpha|y)} = \frac{g(y|\alpha)p(\alpha)}{g(\alpha|y)}. \tag{11}$$

Substituting for $p(\alpha)$ from (11) into (10) gives

$$L(\theta) = L_g(\theta) \int \frac{p(y|\alpha)}{g(y|\alpha)} g(\alpha|y) d\alpha = L_g(\theta) E_g[w(\alpha)], \tag{12}$$

where

$$w(\alpha) \equiv \frac{p(y|\alpha)}{g(y|\alpha)}.$$

One can generate a random sample $\alpha^{(1)}, \ldots, \alpha^{(N)}$ from the density $g(\alpha|y)$ using any of the methods for drawing states in fully Gaussian models. An unbiased Monte Carlo estimate of $L(\theta)$ is

$$\hat{L}_1(\theta) = L_g(\theta)\bar{w}, \tag{13}$$

where $\bar{w} = N^{-1} \sum_{i=1}^{N} w(\alpha^{(i)})$.

It is usually more convenient to work with the log-likelihood, and we can write

$$\log \hat{L}_1(\theta) = \log L_g(\theta) + \log \bar{w}. \tag{14}$$

However, $E[\log \bar{w}] \neq \log E_g[w(\alpha^{(i)}]$, so (14) is a biased estimator of $\log L(\theta)$.

Durbin and Koopman (1997) propose an approximately unbiased estimator of $\log L(\theta)$

given by

$$\log \hat{L}_2(\theta) = \log L_g(\theta) + \log \bar{w} + \frac{s_w^2}{2N\bar{w}^2}, \tag{15}$$

where s_w^2 is an estimator of the variance of the $w(\alpha^{(i)})$ given by

$$s_w^2 = \frac{1}{N-1}\sum_{i=1}^{N}(w(\alpha^{(i)}) - \bar{w})^2.$$

5.1 Modifications to the Algorithm for Approximating $L(\theta)$

We propose here three modifications of the Durbin and Koopman (1997) method for approximating $L(\theta)$. The modified method does not involve Kalman filtering.

First, we use the MMP algorithm to draw α from its conditional distribution given y.

Second, we compute $L_g(\theta)$ as the extreme right hand side of equation (11). The equation holds for any value of α; convenient choices which simplify computations include the prior mean and the posterior mean.

Finally, in the rest of this section we present a method for computing the μ_t and Ξ_t of the fully Gaussian state-space model. It is based on a multivariate normal approximation of $p(\alpha|y)$ at its mode $\hat{\alpha}$ and the application of Result 3.1. It is computationally more efficient than Kalman filtering and smoothing.

We first compute $\hat{\alpha}$ by iterating the following steps until convergence.

1. Using the current value of $\hat{\alpha}$, find the precision $\bar{\bar{H}}$ and co-vector $\bar{\bar{c}}$ of a Gaussian approximation to $p(\alpha|y)$ based on a second-order Taylor expansion of $\log p(\alpha) + \log p(y|\alpha)$ around the point $\hat{\alpha}$.

2. Using the current values of $\bar{\bar{H}}$ and $\bar{\bar{c}}$, compute $\hat{\mu} = \bar{\bar{H}}^{-1}\bar{\bar{c}}$, the mean of the Gaussian approximation, using Result 3.1.

We then use equations (8) and (9) to compute the μ_t and Ξ_t.

We compute the precision $\bar{\bar{H}}$ as $\bar{H} + \tilde{H}$, and the co-vector $\bar{\bar{c}}$ as $\bar{c} + \tilde{c}$, where $\bar{H}$ and $\bar{c}$ are the precision and co-vector of the marginal distribution of α (detailed formulations are provided for our example in the next section), and $\tilde{H}$ and $\tilde{c}$ are the precision and co-vector for a Gaussian density approximating $p(y|\alpha)$ as a function of α up to a multiplicative constant. Since $\tilde{H}$ is block-diagonal and $\bar{H}$ is block-band-diagonal, $\bar{\bar{H}}$ is also block-band-diagonal.

We compute $\tilde{H}$ and $\tilde{c}$ as follows. Let $a(\alpha_t) \equiv -2\log[p(y_t|\alpha_t)]$. We approximate $a(\alpha_t)$ by $\tilde{a}(\alpha_t)$, consisting of the first three terms of the Taylor expansion of $a(\alpha_t)$ around $\hat{\alpha}_t$:

$$a(\alpha_t) \approx \tilde{a}(\alpha_t) = a(\hat{\alpha}_t) + \frac{\partial a(\hat{\alpha}_t)}{\partial \alpha_t}(\alpha_t - \hat{\alpha}_t) + \frac{1}{2}(\alpha_t - \hat{\alpha}_t)' \frac{\partial^2 a(\hat{\alpha}_t)}{\partial \alpha_t \partial \alpha_t'}(\alpha_t - \hat{\alpha}_t).$$

If we complete the square, we obtain

$$\tilde{a}(\alpha_t) = (\alpha_t - h_t^{-1}c_t)' h_t (\alpha_t - h_t^{-1}c_t) + k,$$

where

$$\begin{aligned} h_t &= \frac{1}{2}\frac{\partial^2 a(\hat{\alpha}_t)}{\partial \alpha_t \partial \alpha_t'}, \\ c_t &= h_t \hat{\alpha}_t - \frac{1}{2}\frac{\partial a(\hat{\alpha}_t)}{\partial \alpha_t}, \end{aligned}$$

and k is an unimportant term not depending on α_t. Note that h_t and c_t are the precision and co-vector of a multivariate normal distribution with density proportional to $\exp[-\frac{1}{2}\tilde{a}(\alpha_t)]$.

Since $\log p(y|\alpha)$ is additively separable in the elements of α, it means that it is reasonably well approximated, as a function of α, by $\prod_{t=1}^{n} \exp[-\frac{1}{2}\tilde{a}(\alpha_t)]$, which is proportional

to a multivariate normal distribution with precision $\tilde{H}$ and co-vector $\tilde{c}$, given by

$$\tilde{H} \equiv \begin{bmatrix} h_1 & 0 & \cdots & 0 \\ 0 & h_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_n \end{bmatrix} \quad \text{and} \quad \tilde{c} \equiv \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}.$$

5.2 A Multivariate Poisson Model with Time-Varying Intensities

As an example of a semi-Gaussian state-space model, let us consider a case where $y_t \equiv (y_{t1}, \ldots, y_{tp})$ is a process describing multivariate count data. We assume that the y_t are conditionally independent Poisson given the time-varying stochastic count intensity vector $\lambda_t \equiv (\lambda_{t1}, \ldots, \lambda_{tp})$. Thus the conditional distribution of y_t given λ_t is given by

$$p(y_{t1}, \ldots, y_{tp} | \lambda_{t1}, \ldots, \lambda_{tp}) = \prod_{i=1}^{p} \frac{\exp(-\lambda_{ti}) \lambda_{ti}^{y_{ti}}}{y_{ti}!}, \tag{16}$$

The latent count intensities $\lambda_{t1}, \ldots, \lambda_{tp}$ are assumed to follow

$$\lambda_{ti} = \exp\left(\sum_{j=1}^{m} z_{ij} \alpha_{tj} \right), \quad i = 1, \ldots, n, \tag{17}$$

$$\alpha_{t+1,j} = (1 - \phi_j)\bar{\alpha}_j + \phi_j \alpha_{tj} + \eta_{tj}, \quad j = 1, \ldots, m, \tag{18}$$

where the η_{tj} are independent $N(0, Q_j)$. Denote by Q the diagonal matrix with the Q_j's on the diagonal: $Q = \text{diag}(Q_1, \ldots, Q_m)$. We assume that given the process $\{\eta_t\}$, the y_t are conditionally independent, with conditional probability mass function given by (16). The parameters of the model are $\theta \equiv (\bar{\alpha}_j, \phi_j, Q_j, \gamma, z_{ij})_{i \in \{1,\ldots,p\}, j \in \{1,\ldots,m\}}$.

We now turn to the problem of estimating the likelihood $L(\theta)$ of this particular semi-Gaussian model using the approach of Durbin and Koopman (1997). We first need to

determine the matrix Z_t in the measurement equation (7) of the fully Gaussian model. For cases like this one where the measurement distribution is in the exponential family, they provide a choice for Z_t, which in our case is $Z_t \equiv (z_{ij})_{i=1,\ldots,p;j=1,\ldots,m}$. See Section 4.1 and especially equation (24) in Durbin and Koopman (1997) for details. Also, for this example, the precision $\bar{H}$ and co-vector $\bar{c}$, are given by

$$\bar{H} = \begin{bmatrix} \bar{H}_{11} & \bar{H}_{12} & 0 & \cdots & 0 & 0 \\ \bar{H}_{21} & \bar{H}_{22} & \bar{H}_{23} & \cdots & 0 & 0 \\ 0 & \bar{H}_{32} & \bar{H}_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \bar{H}_{n-1,n-1} & \bar{H}_{n-1,n} \\ 0 & 0 & 0 & \cdots & \bar{H}_{n,n-1} & \bar{H}_{nn} \end{bmatrix}, \quad \bar{c} = \begin{bmatrix} \bar{c}_1 \\ \bar{c}_2 \\ \vdots \\ \bar{c}_{n-1} \\ \bar{c}_n \end{bmatrix}$$

where

$$
\begin{aligned}
\bar{H}_{11} &= \bar{H}_{nn} = Q^{-1}, \\
\bar{H}_{jj} &= \begin{bmatrix} (1+\phi_1^2)/Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (1+\phi_m^2)/Q_m \end{bmatrix}, \quad j = 2, \ldots, n-1, \\
\bar{H}_{j,j+1} &= \bar{H}_{j+1,j} = \begin{bmatrix} -\phi_1/Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -\phi_m/Q_m \end{bmatrix}, \quad j = 1, \ldots, n-1, \\
\bar{c}_1 &= \bar{c}_n = \begin{bmatrix} \bar{\alpha}_1(1-\phi_1)/Q_1 \\ \vdots \\ \bar{\alpha}_m(1-\phi_m)/Q_m \end{bmatrix}, \\
\bar{c}_j &= \begin{bmatrix} \bar{\alpha}_1(1-\phi_1)^2/Q_1 \\ \vdots \\ \bar{\alpha}_m(1-\phi_m)^2/Q_m \end{bmatrix}, \quad j = 2, \ldots, n-1.
\end{aligned}
$$

We compare the computational efficiency of all three methods for estimating the likelihood for this semi-Gaussian state-space model. We do so by counting computational operations and profiling code.

Since a large number of draws from $g(\alpha|y)$ is required for a good approximation of $L(\theta)$, we focus on the marginal computational cost of an additional draw. We will see that for a typical number of draws, the computational overhead associated with the first draw is small.

In the DeJS approach, one additional draw α_t requires the following computations per

observation [see equation (5) of their paper]:

$$\begin{aligned} n_t &= [D_t^{-1} e_t] - K_t^\top r_t, \qquad \epsilon_t = [C_t^{1/2}] N(0, I_p), \qquad \xi_t = \Gamma_t n_t + \epsilon_t, \\ Z\alpha_t &= [y_t - \mu_t] - \xi_t, \qquad r_{t-1} = [Z' D_t^{-1} e_t] + L_t' r_t - [V_t' C_t^{-1}] \epsilon_t. \end{aligned}$$

In the DK approach, one additional draw requires the following computations per observation. (Here we do not simulate α but rather the $Z\alpha_t$, which we obtain more easily by simulating the disturbances ϵ_t according to Section 2.3 of Durbin and Koopman (2002).) There is a forward pass to simulate v_t^+:

$$v_t^+ = \mu_t + Z x_t^+ + \epsilon_t^+, \quad x_{t+1}^+ = T_t x_t^+ + \eta_t^+ - K_t v_t^+,$$

where $\epsilon_t^+ \sim N(0, \Xi_t)$ and $\eta_t^+ \sim N(0, Q)$. This is followed by a backward pass [see equations (4) and (5) and Algorithm 1 of their paper]:

$$\hat{\epsilon}_t^+ = \Xi_t (D_t^{-1} v_t - K_t' r_t), \quad r_{t-1} = [Z' D_t^{-1}] v_t^+ + L_t' r_t,$$

$$\tilde{\epsilon}_t = \hat{\epsilon}_t - \hat{\epsilon}_t^+ + \epsilon_t^+, \quad Z\alpha_t = [y_t - \mu_t] - \tilde{\epsilon}_t,$$

where $\hat{\epsilon}_t$ is pre-computed.

In the MMP approach, one additional draw requires the following computations per observation[1]:

$$\alpha_t = m_t - [\Sigma_t \Omega_{t,t+1}] \alpha_{t+1} + [\Sigma_t^{1/2}] N(0, I_m).$$

The computational costs per observation for an additional draw of α_t are summarized in

[1] Adding $p \times m$ multiplications for each of the $Z\alpha_t$, which are required to evaluate $p(y|\alpha)$.

Table 2: Computational costs per observation per additional draw of α_t

Algorithm	$+/-$	$\times$	$N_{0,1}$
DeJS	$3p + 2m$	$(3p^2 + p)/2 + 2mp + m^2$	p
DK	$6p + 3m$	$(5p^2 + p)/2 + 4mp + 2m + m^2$	$p + m$
MMP	$2m$	$(3m^2 + m)/2 + pm$	m

Table 2.

We profile code for all three methods to see how they perform in practice. We use data on the number of transactions over consecutive two minute intervals for four different stocks in the same industry. For one business day (November 6, 2003), we look at all the transactions for four different gold-mining companies: Agnico-Eagle Mines Limited, Barrick Gold Corporation, Gold Fields Limited and Goldcorp Inc. We use all the transactions recorded during normal trading hours on the New York Stock Exchange Trade and Quote database. This gives 195 observations for each series. The data are plotted in Figure 1.

We take the number of factors to be equal to the number of observed series. That is, $m = p = 4$. To ensure identification, we impose $z_{ii} = 1$ and $z_{ij} = 0$ for $j > i$.

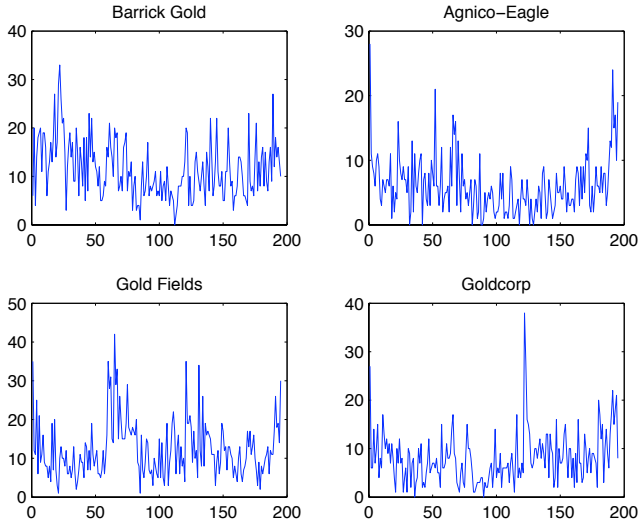
For all three methods, we compute $\hat{\alpha}$ using the fast method presented in Section 5.1. This puts all methods for drawing states on an equal footing. We point out, though, that this gives a small advantage to the estimation of $L(\theta)$ using either the DeJS or DK methods, relative to the case where the Kalman filter and simulation smoother are used to compute $\hat{\alpha}$.

For various values of the size N of the importance sample, Table 3 gives the ratio of the time cost in 100ths of seconds of (i) generating N draws of $\alpha^{(i)}$ and (ii) the total cost of evaluating the log-likelihood once, which consists of the former plus some overhead, including the computation of μ_t, Ξ_t, $\hat{\alpha}$ and the $w(\alpha^{(i)})$. For the latter, we report costs for two different approximations of $\hat{\alpha}$: one using a single iteration of steps 1 and 2 in Section

5.1, the other using five iterations. All times are averaged over 100 replications[2].

First, we see that the cost of evaluating the log-likelihood over and above the cost of drawing the states is around 0.1 second (one iteration for $\hat{\alpha}$) or 0.3 second (five iterations) and that it is the major cost for small number of draws. Second, except for the case $N = 1$, DeJS is computationally more efficient than DK, by a factor of about 2 with $N > 50$. Third, MMP is much more computationally efficient than DeJS and DK for any number of draws, with the efficiency increasing with N. As a point of reference, Durbin and Koopman (1997) consider $N = 200$ (combined with antithetic and control variables) as an acceptable value in an empirical example they consider.

Figure 1: Transactions data



[2]The simulations were performed on an AMD Athlon(tm) 64 X2 5600+ cpu with Matlab R2006a. Note that the reported time costs are in the case where matrix multiplications involving triangular matrices are performed with Matlab's built-in matrix product, which does not take advantage of the triangular structure. We tried dynamically loading a function written in C for triangular matrix multiplication, but the additional overhead exceeded the savings.

Table 3: Time cost of drawing $\alpha^{(i)}$ and the total cost of evaluating the likelihood, as a function of the number of draws N. For the total time cost, numbers are reported when performing one and five iterations to obtain $\hat{\alpha}$. Figures are in 100ths of seconds.

Method		$N = 1$	$N = 10$	$N = 50$	$N = 150$
DeJS	α draw	9.7	22.0	78.3	215.1
	total	(19.2–38.7)	(31.9–51.7)	(89.6–108.9)	(229.2–253.6)
DK	α draw	7.1	34.6	156.6	462.9
	total	(16.7–36.6)	(45.1–64.9)	(168.1–185.7)	(477.8–491.0)
MMP	α draw	4.6	10.0	34.9	103.5
	total	(14.6–33.9)	(20.3–40.4)	(46.5–65.5)	(118.2–136.3)

6 Conclusion

In this paper we introduce a new method for drawing state variables in Gaussian state-space models from their conditional distribution given parameters and observations. The method is quite different from standard methods, such as those of de Jong and Shephard (1995) and Durbin and Koopman (2002), that use Kalman filtering. It is much more in the spirit of Rue (2001), who describes an efficient method for drawing Gaussian random vectors with band diagonal precision matrices. As Rue (2001) recognizes, the distribution $\alpha|y$ in linear Gaussian state-space models is an example.

Our first contribution is computing Ω and c for a widely used and fairly flexible state-space model. These are required inputs for both the CFA of Rue (2001) and the method we described here.

Our second contribution is a new precision-based state smoothing algorithm. It is more computationally efficient for the special case of state-space models, and delivers the conditional means $E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ and conditional variances $\text{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ as a byproduct. These conditional moments turn out to be very useful in an extension of the method, described in McCausland (2008), to non-linear and non-Gaussian state-space models with univariate states.

The algorithm is an extention of a Levinson-like algorithm introduced by Vandebril, Mastronardi, and Van Barel (2007), for solving the equation $Bx = y$, where B is an $n \times n$ symmetric band diagonal matrix and y is a $n \times 1$ vector. The algorithm extends theirs in two ways. First, we modify the algorithm to work with $m \times m$ submatrices of a block band diagonal matrix rather than individual elements of a band diagonal matrix. Second, we use intermediate quantities computed while solving the equation $\Omega\mu = c$ for the mean μ given the precision Ω and co-vector c in order to compute the conditional means $E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ and conditional variances $\mathrm{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$.

Our third contribution is a computational analysis of several state smoothing methods. One can often precompute the Ω_{tt} and $\Omega_{t,t+1}$, in which case the precision-based methods are more efficient than those based on the Kalman filter. The advantage is particularly strong when p is large or when several draws of α are required for each value of the parameters. Kalman filtering, which involves solving systems of p equations in p unknowns, requires $O(p^3)$ scalar multiplications. If the A_t can be pre-computed, or take on only a constant number of values, the precision-based methods require no operations of higher order than p^2, in p. If the Z_t and T_t can also be pre-computed, or take on only a constant number of values, the order drops to p. For large m, our method involves half as many scalar multiplications as CFA.

We consider an applications of our methods to the evaluation of the log-likelihood function for a multivariate Poisson model with latent count intensities.

A Proof of Result 3.1

Suppose $\alpha|y \sim N(\Omega^{-1}c, \Omega^{-1})$ and define

$$\Sigma_1 = (\Omega_{11})^{-1}, \qquad m_1 = \Sigma_1 c_1,$$

$$\Sigma_t = (\Omega_{tt} - \Omega_{t,t-1}\Sigma_{t-1}\Omega_{t-1,t})^{-1}, \qquad m_t = \Sigma_t(c_t - \Omega_{t,t-1}m_{t-1}).$$

Now let $\mu_n \equiv m_n$ and for $t = n-1, \ldots, 1$, let $\mu_t = m_t - \Sigma_t\Omega_{t,t+1}\mu_{t+1}$. Let $\mu = (\mu_1^\top, \ldots, \mu_n^\top)^\top$.

We first show that $\Omega\mu = c$, which means that $\mu = E[\alpha|y]$:

$$\begin{aligned}\Omega_{11}\mu_1 + \Omega_{12}\mu_2 &= \Omega_{11}(m_1 - \Sigma_1\Omega_{12}\mu_2) + \Omega_{12}\mu_2 \\ &= \Omega_{11}((\Omega_{11})^{-1}c_1 - (\Omega_{11})^{-1}\Omega_{12}\mu_2) + \Omega_{12}\mu_2 = c_1.\end{aligned}$$

For $t = 2, \ldots, n-1$,

$$\begin{aligned}&\Omega_{t,t-1}\mu_{t-1} + \Omega_{tt}\mu_t + \Omega_{t,t+1}\mu_{t+1} \\ &= \Omega_{t,t-1}(m_{t-1} - \Sigma_{t-1}\Omega_{t-1,t}\mu_t) + \Omega_{tt}\mu_t + \Omega_{t,t+1}\mu_{t+1} \\ &= \Omega_{t,t-1}m_{t-1} + (\Omega_{tt} - \Omega_{t,t-1}\Sigma_{t-1}\Omega_{t-1,t})\mu_t + \Omega_{t,t+1}\mu_{t+1} \\ &= \Omega_{t,t-1}m_{t-1} + \Sigma_t^{-1}\mu_t + \Omega_{t,t+1}\mu_{t+1} \\ &= \Omega_{t,t-1}m_{t-1} + \Sigma_t^{-1}(m_t - \Sigma_t\Omega_{t,t+1}\mu_{t+1}) + \Omega_{t,t+1}\mu_{t+1} \\ &= \Omega_{t,t-1}m_{t-1} + (c_t - \Omega_{t,t-1}m_{t-1}) = c_t.\end{aligned}$$

$$\begin{aligned}\Omega_{n,n-1}\mu_{n-1} + \Omega_{nn}\mu_n &= \Omega_{n,n-1}(m_{n-1} - \Sigma_{n-1}\Omega_{n-1,n}\mu_n) + \Omega_{nn}\mu_n \\ &= \Omega_{n,n-1}m_{n-1} + \Sigma_n^{-1}\mu_n \\ &= \Omega_{n,n-1}m_{n-1} + \Sigma_n^{-1}m_n \\ &= \Omega_{n,n-1}m_{n-1} + (c_n - \Omega_{n,n-1})m_{n-1} = c_n.\end{aligned}$$

We will now prove that $E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y] = m_t - \Sigma_t\Omega_{t,t+1}\alpha_{t+1}$ and that $\mathrm{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y] =$

Σ_t. We begin with the standard result

$$\alpha_{1:t}|\alpha_{t+1:n}, y \sim \mathrm{N}\left(\mu_{1:t} - \Omega^{-1}_{1:t,1:t}\Omega_{1:t,t+1:n}(\alpha_{t+1:n} - \mu_{t+1:n}), \Omega^{-1}_{1:t,1:t}\right),$$

where μ, α and Ω are partitioned as

$$\mu = \begin{bmatrix} \mu_{1:t} \\ \mu_{t+1:n} \end{bmatrix}, \quad \alpha = \begin{bmatrix} \alpha_{1:t} \\ \alpha_{t+1:n} \end{bmatrix}, \quad \Omega = \begin{bmatrix} \Omega_{1:t,1:t} & \Omega_{1:t,t+1:n} \\ \Omega_{t+1:n,1:t} & \Omega_{t+1:n,t+1:n} \end{bmatrix},$$

with $\mu_{1:t}$, $\alpha_{1:t}$ and $\Omega_{(11)}$ having dimensions $tm \times 1$, $tm \times 1$, and $tm \times tm$ respectively. Note that the only non-zero elements of $\Omega_{(12)}$ come from $\Omega_{t,t+1}$. We can therefore write the univariate conditional distribution $\alpha_t|\alpha_{t+1:n}$ as

$$\alpha_t|\alpha_{t+1:n} \sim \mathrm{N}(\mu_t - (\Omega^{-1}_{1:t,1:t})_{tt}\Omega_{t,t+1}(\alpha_{t+1} - \mu_{t+1}), (\Omega^{-1}_{1:t,1:t})_{tt}).$$

The following inductive proof establishes the result $\mathrm{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y] = \Sigma_t$:

$$(\Omega_{11})^{-1} = \Sigma_1$$

$$\begin{aligned} (\Omega^{-1}_{1:t,1:t})_{tt} &= (\Omega_{tt} - \Omega_{t,1:t-1}\Omega^{-1}_{1:t-1,1:t-1}\Omega_{1:t-1,t})^{-1} \\ &= (\Omega_{tt} - \Omega_{t,t-1}\Sigma_{t-1}\Omega_{t-1,t})^{-1} = \Sigma_t. \end{aligned}$$

As for the conditional mean,

$$E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y] = \begin{cases} \mu_t - \Sigma_t\Omega_{t,t+1}(\alpha_{t+1} - \mu_{t+1}) & t = 1, \ldots, n-1 \\ \mu_n & t = n. \end{cases}$$

By the definition of μ_t, $m_t = \mu_t + \Sigma_t \Omega_{t,t+1} \mu_{t+1}$, so we obtain

$$E[\alpha_t | \alpha_{t+1}, \ldots, \alpha_n, y] = \begin{cases} m_t - \Sigma_t \Omega_{t,t+1} \alpha_{t+1} & t = 1, \ldots, n-1 \\ m_n & t = n. \end{cases}$$

References

Boivin, J., and Giannoni, M. (2006). 'DSGE Models in a Data-Rich Environment', Working Paper 12772, National Bureau of Economic Research.

Carter, C. K., and Kohn, R. (1994). 'On Gibbs Sampling for State Space Models', *Biometrika*, 81(3): 541–553.

——— (1996). 'Markov chain Monte Carlo in conditionally Gaussian State Space Models', *Biometrika*, 83(3): 589–601.

de Jong, P., and Shephard, N. (1995). 'The Simulation Smoother for Time Series Models', *Biometrika*, 82(1): 339–350.

Durbin, J., and Koopman, S. J. (1997). 'Monte Carlo maximum likelihood estimation for non-Gaussian state space models', *Biometrika*, 84(3): 669–684.

——— (2002). 'A Simple and Efficient Simulation Smoother for State Space Time Series Analysis', *Biometrika*, 89(3): 603–615.

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2000). 'The Generalized Dynamic Factor Model: Identification and Estimation', *Review of Economics and Statistics*, 82(4): 540–554.

Frühwirth-Schnatter, S. (1994). 'Data augmentation and Dynamic Linear Models', *Journal of Time Series Analysis*, 15: 183–202.

Frühwirth-Schnatter, S., and Wagner, H. (2006). 'Auxiliary mixture sampling for parameter-driven models of time series of counts with applications to state space modelling', *Biometrika*, 93: 827–841.

Kim, S., Shephard, N., and Chib, S. (1998). 'Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models', *Review of Economic Studies*, 65(3): 361–393.

Knorr-Held, L., and Rue, H. (2002). 'On Block Updating in Markov Random Field Models for Disease Mapping', *Scandinavian Journal of Statistics*, 29: 597–614.

McCausland, W. J. (2008). 'The HESSIAN Method (Highly Efficient State Smoothing, In A Nutshell)', Cahiers de recherche du Département de sciences économiques, Université de Montréal, no. 2008-03.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in C*. Cambridge University Press, Cambridge, second edn., The art of scientific computing.

Rue, H. (2001). 'Fast Sampling of Gaussian Markov Random Fields with Applications', *Journal of the Royal Statistical Society Series B*, 63: 325–338.

Shephard, N., and Pitt, M. K. (1997). 'Likelihood Analysis of Non-Gaussian Measurement Time Series', *Biometrika*, 84(3): 653–667.

Stock, J. H., and Watson, M. W. (1999). 'Forecasting Inflation', *Journal of Monetary Economics*, 44: 293–335.

——— (2002). 'Macroeconomic Forecasting Using Diffusion Indexes', *Journal of Business and Economic Statistics*, 20: 147–162.

Stroud, J. R., Müller, P., and Polson, N. G. (2003). 'Nonlinear State-Space Models With State-Dependent Variances', *Journal of the American Statistical Association*, 98: 377–386.

Vandebril, R., Mastronardi, N., and Van Barel, M. (2007). 'A Levinson-like algorithm for symmetric strongly nonsingular higher order semiseparable plus band matrices', *Journal of Computational and Applied Mathematics*, 198(1): 74–97.