

# 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: October 6, 2009

## 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](http://www.cirano.qc.ca/~mccauslw).

†Same mailing address as McCausland. e-mail: [shirley.miller.lira@umontreal.ca](mailto: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](mailto: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 convenient but 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 for importance sampling, as Durbin and Koopman (1997) do to compute likelihood functions, or as a proposal distribution in

a Metropolis-Hastings update, as Shephard and Pitt (1997) do for posterior Markov chain Monte Carlo simulation.

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

$$y_t = X_t\beta + Z_t\alpha_t + G_tu_t, \quad t = 1, \dots, n, \quad (1)$$

$$\alpha_{t+1} = W_t\beta + T_t\alpha_t + H_tu_t, \quad t = 1, \dots, n-1, \quad (2)$$

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

where  $y_t$  is a  $p \times 1$  vector of dependent variables,  $\alpha_t$  is a  $m \times 1$  vector of state variables, and  $\beta$  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  $y \equiv (y_1^\top, \dots, y_n^\top)^\top$  and  $\alpha \equiv (\alpha_1^\top, \dots, \alpha_n^\top)^\top$ .

We will consider the familiar and important question of simulation smoothing, which is drawing  $\alpha$  as a block from its conditional distribution given  $y$ . This is an important component of various sampling methods for learning about the posterior distribution of states, parameters and other functions of interest. We focus on Gaussian linear models, and recall that simulation smoothing for these models is a component of simulation smoothing for more general models they approximate.

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) based on the Cholesky decomposition of the precision (inverse of variance) of the random field. He also recognizes that the conditional distribution of  $\alpha$  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.

One uses the Kalman filter not only for simulation smoothing, but also to evaluate the likelihood function for Gaussian linear state-space models. We can do the same using the CFA and our method. Both give evaluations of  $f(\alpha|y)$  for arbitrary  $\alpha$  with little additional computation. We can then evaluate the likelihood as

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

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

The Kalman filter also delivers intermediate quantities that are useful for computing filtering distributions, the conditional distributions of  $\alpha_1, \dots, \alpha_t$  given  $y_1, \dots, y_t$ , for various values of  $t$ . While it is difficult to use the CFA algorithm to compute these distributions efficiently, it is fairly straightforward to do so using our method.

We make four main contributions in this paper. The first is a new method, outlined in Section 2, for drawing states in state-space models. Like the CFA, it uses the precision and covector (precision times mean) of the conditional distribution of  $\alpha$  given  $y$  and does not use the Kalman filter. Unlike the CFA, it generates the conditional means  $E[\alpha_t|\alpha_{t+1}, \dots, \alpha_n, y]$  and conditional variances  $\text{Var}[\alpha_t|\alpha_{t+1}, \dots, \alpha_n, y]$  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. With little additional computation, one can also compute the conditional means  $E[\alpha_t|y_1, \dots, y_t]$

and variances  $\text{Var}[\alpha_t|y_1, \dots, y_t]$ , which together specify the filtering distributions, useful for sequential learning.

The second main contribution, described in Section 3, is a careful analysis of 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  $\alpha$  are required. For the important special case of state-space models, our new method is up to 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  $\alpha$  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).

Our third contribution is illustrate these simulation smoothing methods using an empirical application. In Section 5, we use them to approximating the likelihood function for a multivariate Poisson state space using importance sampling. Latent states govern time-varying intensities and observed data are transaction counts in financial markets.

The final contribution is the explicit derivation, in Appendix A, of the precision and covector of the conditional distribution of  $\alpha$  given  $y$  in Gaussian linear state-space models. These two objects are the inputs to the CFA and our new method.

We conclude in Section 6.

## 2 Precision-Based Methods for Simulation Smoothing

In this section we discuss two methods for state smoothing using the precision  $\Omega$  and covector  $c$  of the conditional distribution of  $\alpha$  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.

We will take  $\Omega$  and  $c$  as given here. In Appendix A, we show how to compute  $\Omega$  and  $c$  in terms of  $X_t, Z_t, G_t, W_t, T_t, H_t, a_1$  and  $P_1$ , assuming that the stacked innovation  $((G_t u_t)^\top, (H_t u_t)^\top)^\top$  has full rank.

The full rank condition is frequently, but not always, satisfied 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 pointed out by Rue (2001).

First, we can draw  $\alpha$  conditional on the linear equality restriction  $A\alpha + b$  by drawing  $\tilde{\alpha}$  unconditionally and then “conditioning by Kriging” to obtain  $\alpha$ . This gives  $\alpha = \tilde{\alpha} - \Omega^{-1}A^\top(A\Omega^{-1}A^\top)^{-1}(A\tilde{\alpha} + b)$ . One can precompute the columns of  $\Omega^{-1}A^\top$  in the same way as we compute  $\mu = \Omega^{-1}c$  in Appendix B, then precompute  $A\Omega^{-1}A^\top$  and  $-\Omega^{-1}A^\top(A\Omega^{-1}A^\top)^{-1}$ .

Second, state-space models where the innovation has less than full rank are usually more naturally expressed in another form, one that allows application of the CFA method. Take for example a model where a univariate latent variable  $\alpha_t$  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  $\alpha$  given  $y$  is a GMRF and one can apply the CFA method directly.

Rue (2001) introduces a simple procedure for drawing a Gaussian random vector  $\alpha$  with a band-diagonal precision matrix  $\Omega$ . We let  $N$  be the length of  $\alpha$  and  $b$  be the number of non-zero subdiagonals. By symmetry, the bandwidth of  $\Omega$  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  $\alpha^*$ , where  $\epsilon \sim N(0, I_N)$ , using band back-substitution. Then  $\alpha^* + \mu$ , where  $\mu$  is the mean of  $\alpha$ , is a draw from the distribution of  $\alpha$ . If the covector  $c$  of  $\alpha$  is readily available but not  $\mu$ , one can pre-compute  $L^{-1}c$  using band back-substitution, then compute the draw  $\alpha^*$  by solving the equation  $L^\top \alpha^* = L^{-1}c + \epsilon$  for  $\alpha^*$ , also using band back-substitution. Rue (2001) recognizes that the vector of states  $\alpha$  in Gaussian linear state-space models is an example of a Gaussian Markov random fields. Appendix A explicitly derives  $\Omega$  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  $\alpha$  based on the precision and covector of its conditional distribution of  $\alpha$  given  $y$ . We draw the  $\alpha_t$  backwards, each  $\alpha_t$  from the distribution  $\alpha_t | \alpha_{t+1}, \dots, \alpha_n, y$ . The following result, proved in Appendix B, allows us to draw  $\alpha$  and evaluate  $E[\alpha | y]$  in time  $n$ .

**Result 2.1** *If  $\alpha | y \sim N(\Omega^{-1}c, \Omega^{-1})$ , where  $\Omega$  has the block band structure of equation (13), then*

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

where

$$\begin{aligned} \Sigma_1 &= (\Omega_{11})^{-1}, & m_1 &= \Sigma_1 c_1, \\ \Sigma_t &= (\Omega_{tt} - \Omega_{t-1,t}^\top \Sigma_{t-1} \Omega_{t-1,t})^{-1}, & m_t &= \Sigma_t (c_t - \Omega_{t-1,t}^\top m_{t-1}), \end{aligned}$$

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

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 2.1 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  $E[\alpha_t|\alpha_{t+1}, \dots, \alpha_n, y]$  and  $\text{Var}[\alpha_t|\alpha_{t+1}, \dots, \alpha_n, y]$ .

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

1. Compute  $\Sigma_1 = \Omega_{11}^{-1}$ ,  $m_1 = \Sigma_1 c_1$ .
2. For  $t = 2, \dots, n$ , compute  $\Sigma_t = (\Omega_{tt} - \Omega_{t-1,t}^\top \Sigma_{t-1} \Omega_{t-1,t})^{-1}$ ,  $m_t = \Sigma_t (c_t - \Omega_{t-1,t}^\top m_{t-1})$ .
3. Draw  $\alpha_n \sim N(m_n, \Sigma_n)$ .
4. For  $t = n - 1, \dots, 1$ , draw  $\alpha_t \sim N(m_t - \Sigma_t \Omega_{t,t+1} \alpha_{t+1}, \Sigma_t)$ .

We consider now the problem of computing the filtering distribution at time  $t$ , the conditional distribution of  $\alpha_t$  given  $y_1, \dots, y_t$ . Since  $\alpha$  and  $y$  are jointly multivariate Gaussian, this distribution is also Gaussian and it is enough to compute the mean  $E[\alpha_t|y_1, \dots, y_t]$  and variance  $\text{Var}[\alpha_t|y_1, \dots, y_t]$ . It turns out we can do this with very little additional computation.

Fix  $t$  and consider the two cases  $n = t$  and  $n > t$ . It is easy to see (in Appendix A) that for  $\tau = 1, \dots, t - 1$ , the values of  $c_\tau$ ,  $\Omega_{\tau\tau}$  and  $\Omega_{\tau,\tau+1}$  do not differ from case to case. Therefore the values of  $m_\tau$  and  $\Sigma_\tau$  do not vary from case to case either. We can use

equation (14) (for  $\Omega_{nn}$ , taking  $n = t$ ) to compute

$$\tilde{\Omega}_{tt} \equiv Z_t^\top (G_t G_t^\top)^{-1} Z_t + A_{22,t-1},$$

and equation (15) (for  $c_{nn}$ , taking  $n = t$ ) to compute

$$\tilde{c}_t \equiv Z_t^\top (G_t G_t^\top)^{-1} (y_t - X_t \beta) - A_{21,t-1} (y_{t-1} - X_{t-1} \beta) + A_{22,t-1} (W_{t-1} \beta).$$

Then

$$\text{Var}[\alpha_t | y_1, \dots, y_t] = \tilde{\Sigma}_t \equiv (\tilde{\Omega}_{tt} - \Omega_{t-1,t}^\top \Sigma_{t-1} \Omega_{t-1,t})^{-1}$$

and

$$E[\alpha_t | y_1, \dots, y_t] = \tilde{m}_t \equiv \tilde{\Sigma}_t (\tilde{c}_t - \Omega_{t-1,t}^\top m_{t-1}).$$

### 3 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.

### 3.1 Fixed Costs

We first consider the cost of computing the precision  $\Omega$  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  $\Omega_{tt}$  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 (13) 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  $\Omega$ , and the computation of the  $\Sigma_t$  and  $m_t$  of Result 2.1. 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.

### 3.2 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 four 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 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.

Table 1: Scalar multiplications needed for pre-computation.

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

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  $\eta$  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  $\alpha$  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  $\epsilon_t$  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,  $\alpha_{t+1}$  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  $\Lambda_t$  (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  $\Lambda_t$  once and convert the divisions into multipli-

cations, 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  $\Omega$  has elements equal to zero in its first  $2m - 1$  subdiagonals.

## 4 An Experiment with Artificial Data

Regression model with time-varying regression parameters:

$$\beta_{t+1} = \beta_t + w_t, \quad w_t \sim N(0, Q),$$

$$y_t = x_t \beta_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma_\epsilon^2),$$

where  $\beta_t$  is a time-varying  $m$ -vector of regression coefficient,  $x_t$  is a vector of observed regressors and  $y_t$  is a univariate observed dependent variable.

We take  $\beta_1 = (1, -0.5, 0.2, 0.1)$ ,  $Q = \text{diag}(0.1, 0.05, 0.01, 0.02)$  and  $\sigma_\epsilon^2 = 0.05$ .  $x_{t1} = 1$  and  $x_{ti} \sim N(1, 1)$  for  $i \neq 1$ .

## 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  $\theta$ , the vector of parameters.

The approach is as follows. The likelihood function  $L(\theta)$  we wish to evaluate is

$$L(\theta) = p(y) = \int p(y, \alpha) d\alpha = \int p(y|\alpha)p(\alpha) d\alpha. \quad (4)$$

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

$$L_g(\theta) = g(y) = \frac{g(y, \alpha)}{g(\alpha|y)} = \frac{g(y|\alpha)p(\alpha)}{g(\alpha|y)}. \quad (5)$$

Substituting for  $p(\alpha)$  from (5) into (4) gives

$$L(\theta) = L_g(\theta) \int \frac{p(y|\alpha)}{g(y|\alpha)} g(\alpha|y) d\alpha = L_g(\theta) E_g[w(\alpha)], \quad (6)$$

where

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

One can generate a random sample  $\alpha^{(1)}, \dots, \alpha^{(N_s)}$  from the density  $g(\alpha|y)$  using any of the methods for drawing states in fully Gaussian models.

The approximating state-space model has the following form:

$$y_t = \mu_t + Z\alpha_t + \epsilon_t, \quad (7)$$

where the  $\epsilon_t$  are independent  $N(0, \Xi_t)$  and independent of the state equation innovations. 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  $\alpha$ , evaluated at  $\hat{\alpha} = E_g[\alpha|y]$ , where  $E_g$  denotes expectation with respect to the Gaussian density  $g(\alpha|y)$ . Durbin and Koopman (1997) use routine Kalman filtering and smoothing to find  $\hat{\alpha}$ .

## 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  $\alpha$  from its conditional distribution given  $y$ .

Second, we compute  $L_g(\theta)$  as the extreme right hand side of equation (5). The equation holds for any value of  $\alpha$ ; 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  $\hat{\alpha}$  based on a multivariate normal approximation of  $p(\alpha|y)$  at its mode  $\hat{\alpha}$  and the application of Result 2.1. We do so by iterating the following steps until convergence.

1. Using the current value of  $\hat{\alpha}$ , find the precision  $\bar{\bar{H}}$  and co-vector  $\bar{c}$  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  $\bar{c}$ , compute  $\hat{\mu} = \bar{\bar{H}}^{-1}\bar{c}$ , the mean of the Gaussian approximation, using Result 2.1.

We compute the precision  $\bar{\bar{H}}$  as  $\bar{H} + \tilde{H}$ , and the co-vector  $\bar{c}$  as  $\bar{c} + \tilde{c}$ , where  $\bar{H}$  and  $\bar{c}$  are the precision and co-vector of the marginal distribution of  $\alpha$  (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  $\alpha$  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

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

and  $k$  is an unimportant term not depending on  $\alpha_t$ . 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  $\alpha$ , it means that it is reasonably well approximated, as a function of  $\alpha$ , 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  $y_t \equiv (y_{t1}, \dots, y_{tp})$  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

$\lambda_t \equiv (\lambda_{t1}, \dots, \lambda_{tp})$ . Thus the conditional distribution of  $y_t$  given  $\lambda_t$  is given by

$$p(y_{t1}, \dots, y_{tp} | \lambda_{t1}, \dots, \lambda_{tp}) = \prod_{i=1}^p \frac{\exp(-\lambda_{ti}) \lambda_{ti}^{y_{ti}}}{y_{ti}!}, \quad (8)$$

The latent count intensities  $\lambda_{t1}, \dots, \lambda_{tp}$  are assumed to follow a factor model:

$$\lambda_{ti} = \exp \left( \sum_{j=1}^m z_{ij} \alpha_{tj} \right), \quad i = 1, \dots, n, \quad (9)$$

$$\alpha_{t+1,j} = (1 - \phi_j) \bar{\alpha}_j + \phi_j \alpha_{tj} + \eta_{tj}, \quad j = 1, \dots, m, \quad (10)$$

where the  $\eta_{tj}$  are independent  $N(0, Q_j)$ . Denote by  $Q$  the diagonal matrix with the  $Q_j$ 's on the diagonal:  $Q = \text{diag}(Q_1, \dots, Q_m)$ . We assume that given the process  $\{\eta_t\}$ , the  $y_t$  are conditionally independent, with conditional probability mass function given by (8). The parameters of the model are  $\theta \equiv (\bar{\alpha}_j, \phi_j, Q_j, z_{ij})_{i \in \{1, \dots, p\}, j \in \{1, \dots, m\}}$ . To ensure identification<sup>1</sup>, we impose  $z_{ii} = 1$  and  $z_{ij} = 0$  for  $j > i$ . We also assume that  $\alpha_{1,j} \sim N(\bar{\alpha}_j, Q_j / (1 - \phi_j^2))$ .

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). For this example, the precision  $\bar{H}$  and co-vector  $\bar{c}$ , are given by

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

---

<sup>1</sup>See for example Heaton and Solo (2004).

where

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

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, the computational overhead associated with the first draw being small. For all four methods, we compute  $\hat{\alpha}$  using the fast method presented in Section 5.1. For both MMP and CFA, we add  $p \times m$  multiplications for each of the  $Z\alpha_t$ , which are required to evaluate  $p(y|\alpha)$ . The computational costs per observation for an additional draw of  $\alpha_t$  are summarized in Table 2.

Table 2: Computational costs per observation per additional draw of  $\alpha_t$

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

We profile code for all four 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.

For the case where the number of factors is equal to the number of series, that is  $m = p = 4$ , and for various values of the size  $N_s$  of the importance sample, Table 3 gives the time cost in 100ths of seconds of generating  $N_s$  draws of  $\alpha^{(i)}$ . All times are averaged over 100 replications<sup>2</sup>. We report two series of results for the MMP algorithm. A first series for the timing results for a Matlab-only implementation of the MMP algorithm and a second series where the precomputation (steps 1 and 2 of the algorithm) and draws (steps 3 and 4) are coded in C. The latter gives a better comparison with the Matlab implementation of CFA that is able to use specialized libraries to compute the Choleski decomposition and perform the band back-substitution<sup>3</sup>.

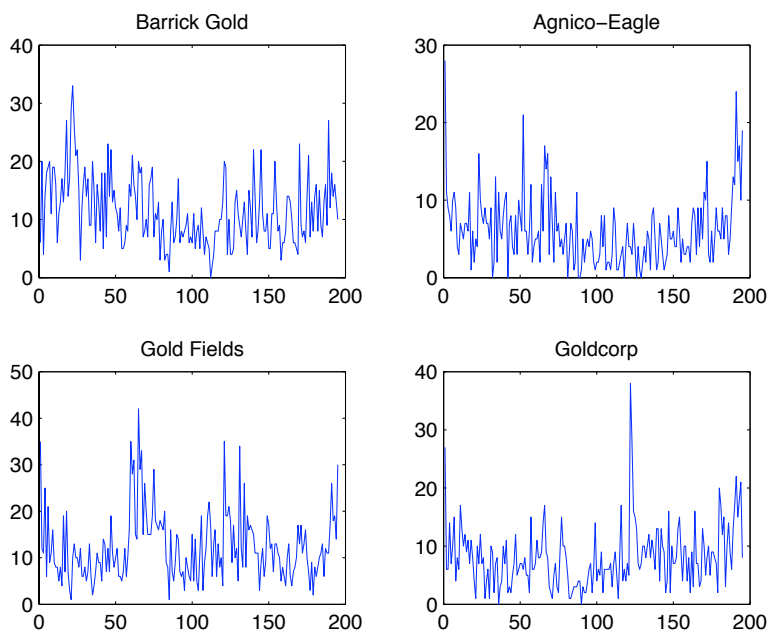
First, we see that for a single draw, DK is computationally more efficient than DeJS and MMP (Matlab) but as the number of draws increase, there is a value of  $N_s$  after

<sup>2</sup>The simulations were performed on a MacBook Pro 2.4 GHz with Matlab R2008b.

<sup>3</sup>By declaring  $\Omega$  to be a sparse matrix, Matlab can use cholmod, a sparse Cholesky factorization package.

which it is the opposite. This point is reached more quickly for MMP (Matlab) than for DeJS. Second, these first three methods are dominated for every value of  $N_s$  by the CFA algorithm. This is the result of requiring less operations (compared to DeJS and DK) and being very efficiently implemented in Matlab (no interpreted loop over  $t$ ). Third, we see that MMP (C) is numerically more efficient than CFA. One can see for example that in the time it takes to obtain 50 draws of  $\alpha^{(i)}$  with CFA, we can generate 250 such draws with MMP (C). As a point of reference, Durbin and Koopman (1997) consider  $N_s = 200$  (combined with antithetic and control variables) as an acceptable value in an empirical example they consider.

Figure 1: Transactions data



We next discuss the results of the estimation of this multivariate count data model. The

Table 3: Time cost of drawing  $\alpha^{(i)}$  as a function of the number of draws  $N_s$ . Figures are in 100ths of seconds.

Method	$N_s = 1$	$N_s = 10$	$N_s = 50$	$N_s = 150$	$N_s = 250$
DeJS	9.27	9.71	10.33	12.65	14.06
DK	6.46	7.46	8.79	13.76	16.88
MMP (Matlab)	6.60	6.86	7.17	8.11	8.96
CFA	2.20	2.36	2.59	3.42	4.46
MMP (C)	1.33	1.42	1.62	2.15	2.56

estimates, standard errors<sup>4</sup> and log-likelihood values are presented in Table 4 for different values of  $m$ , the number of latent factors. These results are obtained with  $N_s = 500$  and antithetic variables. To select a value for  $m$  we can't use a test statistic such as the likelihood ratio test with the usual  $\chi^2$  limit distribution. For example, a likelihood ratio test for  $m = 1$  versus  $m = 2$  where we test  $z_{32} = z_{42} = 0$  leaves the parameters  $\bar{\alpha}_2$ ,  $\phi_2$  and  $Q_2$  unidentified under the null. An alternative is to use an information criterion such as  $AIC = -2 \log L(\theta) + 2 \dim(\theta)$  and  $SIC = -2 \log L(\theta) + \log(pn) \dim(\theta)$ . See Song and Belin (2008) for an example. These two criteria both suggest that  $m$  should be equal to four. For the model with  $m = 4$ , we can see that the first factor is the more persistent with  $\hat{\phi}_1 = 0.7710$ , the other all being closer to zero. It is also the factor with innovations with the highest variance and with with the highest corresponding factor loadings, the three biggest  $z$ 's being  $z_{21}$ ,  $z_{31}$  and  $z_{41}$ .

## 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

<sup>4</sup>See Durbin and Koopman (2001, Chapter 12).

Table 4: Estimation results for the model for different values of  $m$ . The standard errors are between parantheses.

	$m = 1$		$m = 2$		$m = 3$		$m = 4$	
$\bar{\alpha}_1$	2.0569	(0.0101)	2.0207	(0.0050)	1.9999	(0.0140)	2.0013	(0.0049)
$\bar{\alpha}_2$			0.4022	(0.0718)	0.7311	(0.2105)	0.7004	(0.0821)
$\bar{\alpha}_3$					0.5360	(0.0218)	0.2939	(0.0184)
$\bar{\alpha}_4$							0.3858	(0.0489)
$\phi_1$	0.7873	(0.0007)	0.7402	(0.0255)	0.7595	(0.0079)	0.7710	(0.0059)
$\phi_2$			0.1780	(0.0549)	0.1233	(0.0353)	0.2829	(0.0144)
$\phi_3$					0.0886	(0.0162)	0.0412	(0.0020)
$\phi_4$							0.1182	(0.0025)
$Q_1$	0.1582	(0.0019)	0.2225	(0.0207)	0.2250	(0.0041)	0.2142	(0.0077)
$Q_2$			0.0321	(0.0030)	0.1316	(0.0628)	0.1378	(0.0133)
$Q_3$					0.1451	(0.0262)	0.1678	(0.0074)
$Q_4$							0.1405	(0.0120)
$z_{21}$	0.9928	(0.0064)	0.8170	(0.0358)	0.6626	(0.1000)	0.6714	(0.0388)
$z_{31}$	0.9965	(0.0065)	0.5830	(0.1243)	0.6449	(0.0380)	0.6738	(0.0242)
$z_{41}$	0.9882	(0.0065)	0.6073	(0.1192)	0.5785	(0.0148)	0.6098	(0.0359)
$z_{32}$			2.2228	(0.2558)	0.3012	(0.0665)	0.5203	(0.0078)
$z_{42}$			2.0586	(0.2555)	0.6674	(0.4069)	0.4394	(0.0247)
$z_{43}$					0.7747	(0.3121)	0.3856	(0.0105)
$\log L(\hat{\theta})$	-2432.71		-2378.44		-2350.67		-2324.22	

(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  $\Omega$  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  $E[\alpha_t|\alpha_{t+1}, \dots, \alpha_n, y]$  and conditional variances  $\text{Var}[\alpha_t|\alpha_{t+1}, \dots, \alpha_n, y]$  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 extension 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  $\mu$  given the precision  $\Omega$  and co-vector  $c$  in order to compute the conditional means  $E[\alpha_t|\alpha_{t+1}, \dots, \alpha_n, y]$  and conditional variances  $\text{Var}[\alpha_t|\alpha_{t+1}, \dots, \alpha_n, y]$ .

Our third contribution is a computational analysis of several state smoothing methods. One can often precompute the  $\Omega_{tt}$  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  $\alpha$  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 Derivation of $\Omega$ and $c$

Here we derive expressions for the precision  $\Omega$  and covector  $c$  of the conditional distribution of  $\alpha$  given  $y$ , for the Gaussian linear state-space model described in equations (1), (2) and (3). The matrix  $\Omega$  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.

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.

Clearly  $\alpha$  and  $y$  are jointly Gaussian and therefore the conditional distribution of  $\alpha$  given  $y$  is also Gaussian. We can write the log conditional density of  $\alpha$  given  $y$  as

$$\log f(\alpha|y) = -\frac{1}{2} \left[ \alpha^\top \Omega \alpha - 2c^\top \alpha \right] + k, \quad (11)$$

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

$$\log f(\alpha|y) = \log f(\alpha, y) - \log f(y) = -\frac{1}{2} g(\alpha, y) + k', \quad (12)$$

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  $\alpha$ .

Matching linear and quadratic terms in the  $\alpha_t$  between equations (11) and (12), we obtain

$$\Omega \equiv \begin{bmatrix} \Omega_{11} & \Omega_{12} & 0 & \dots & 0 \\ \Omega_{12}^\top & \Omega_{22} & \Omega_{23} & \ddots & \vdots \\ 0 & \Omega_{23}^\top & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \Omega_{n-1,n-1} & \Omega_{n-1,n} \\ 0 & \dots & 0 & \Omega_{n-1,n}^\top & \Omega_{nn} \end{bmatrix} \quad c \equiv \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \quad (13)$$

where

$$\begin{aligned}
\Omega_{11} &\equiv Z_1^\top A_{11,1} Z_1 + Z_1^\top A_{12,1} T_1 + T_1^\top A_{21,1} Z_1 + T_1^\top A_{22,1} T_1 + P_1^{-1}, \\
\Omega_{tt} &\equiv Z_t^\top A_{11,t} Z_t + Z_t^\top A_{12,t} T_t + T_t^\top A_{21,t} Z_t + T_t^\top A_{22,t} T_t + A_{22,t-1}, \quad t = 2, \dots, n-1, \\
\Omega_{nn} &\equiv Z_n^\top (G_n G_n^\top)^{-1} Z_n + A_{22,n-1}, \\
\Omega_{t,t+1} &\equiv -Z_t^\top A_{12,t} - T_t^\top A_{22,t}, \quad t = 1, \dots, n-1,
\end{aligned} \tag{14}$$

$$c_1 \equiv (Z_1^\top A_{11,1} + T_1^\top A_{21,1})(y_1 - X_1 \beta) - (Z_1^\top A_{12,1} + T_1^\top A_{22,1})(W_1 \beta) + P_1^{-1} a_1,$$

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

$$c_n \equiv Z_n^\top (G_n G_n^\top)^{-1} (y_n - X_n \beta) - A_{21,n-1}(y_{n-1} - X_{n-1} \beta) + A_{22,n-1}(W_{n-1} \beta). \tag{15}$$

## B Proof of Result 2.1

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

$$\Sigma_1 = \Omega_{11}^{-1}, \quad m_1 = \Sigma_1 c_1,$$

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

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

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  $t = 2, \dots, n-1$ ,

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

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

We will now prove that  $E[\alpha_t | \alpha_{t+1}, \dots, \alpha_n, y] = m_t - \Sigma_t \Omega_{t,t+1} \alpha_{t+1}$  and that  $\text{Var}[\alpha_t | \alpha_{t+1}, \dots, \alpha_n, y] = \Sigma_t$ . We begin with the standard result

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

where  $\mu$ ,  $\alpha$  and  $\Omega$  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

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

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

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

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

As for the conditional mean,

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

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

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

## 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.
- (2001). *Time series analysis by state space methods*, vol. 24 of *Oxford Statistical Science Series*. Oxford University Press, Oxford.
- (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.
- Heaton, C., and Solo, V. (2004). ‘Identification of causal factor models of stationary time series’, *Econometrics Journal*, 7(2): 618–627.

- 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.
- Song, J., and Belin, T. R. (2008). ‘Choosing an appropriate number of factors in factor analysis with incomplete data’, *Computational Statistics and Data Analysis*, 52(7): 3560–3569.
- 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.