Efficient Matrix Approach for Classical Inference in State Space Models*

Davide Delle Monache†
Bank of Italy

Ivan Petrella‡
University of Warwick & CEPR

January 10, 2019

Abstract

In this work we explore a novel approach to estimating Gaussian state space models in the classical framework without making use of the Kalman filter and Kalman smoother. By formulating the model in matrix form, we obtain expressions for the likelihood function and the smoothed state vector that are computationally feasible and generally more efficient than the standard filtering approach. Finally, we highlight a convenient way to retrieve the filtering weights and to deal with data irregularities.

JEL codes: C22, C32, C51, C53, C82.

Keywords: State space models, likelihood, smoother, sparse matrices.

*The views expressed in this paper are those of the authors and do not necessarily reflect those of the Bank of Italy.

†Bank of Italy. Economic Outlook and Monetary Policy Directorate. Email: davide.dellemonache@bancaditalia.it

‡Warwick Business School, University of Warwick and CEPR. Email: Ivan.Petrella@wbs.ac.uk

1 Introduction

State space models have a long history in time series econometrics and, owing to their versatility, they are nowadays ubiquitous in economics and finance. In this setting, the Kalman filter (KF) is the main tool to calculate recursively the likelihood function, whereas the Kalman smoother (KS) recursively retrieves the state vector given all the available observations. Harvey (1989) provides a comprehensive treatment of state space methods in the frequentist framework, while Durbin and Koopman (2012) provide an extensive treatment for both the classical and Bayesian approaches.

In this note we illustrate how to estimate Gaussian state space models in the classical framework without using the KF and KS. By taking advantage of the matrix representation, we derived closed form expressions for the likelihood function and the smoothed state vector that are computationally feasible. Although the matrix formulation is not new in the literature, this approach has been typically considered to be unfeasible and inefficient compared with the recursive approach based on the KF and KS (see Durbin and Koopman, 2012, sec. 4.13). One of the main purposes of this note is to highlight how the matrix formulation is not only tractable but can also be computationally more efficient than the recursive approach. In particular, for large systems the matrix approach is order of magnitude more efficient than the standard approach.

Our work draws on Chan and Jeliazkov (2009) and McCausland et al. (2011), who propose an efficient precision-based method to simulate the state vector in the Bayesian framework. In our paper, we map their findings to the classical framework and highlight how similar computational gains exist not only for the estimation of the state vector but also for evaluating the likelihood function, thereby rendering the matrix approach feasible for the maximum likelihood estimation (MLE). Moreover, we show how to recover the weighting function that maps the observations to the state vector (see Koopman and Harvey, 2003), and how to deal with the presence of missing observations in the data (see Harvey and Pierce, 1984). The latter extension is particularly important from a practical perspective, as the flexibility of the KF in dealing with missing observations is one of the key advantages of state space methods. For instance, it makes it possible to deal with mixed frequency (Mariano and Murasawa, 2003) and the 'ragged edge' in the data that is typically present in real-time nowcasting applications (Giannone, Reichlin and Small, 2008). These extensions are novel and potentially of interest in a Bayesian setting as a complement to the findings of Chan and Jeliazkov (2009) and McCausland et al. (2011).

The rest of the note is organized as follows. In Section 2 we present an efficient implementation of the matrix approach. Section 3 presents additional results on the weighting function and the case of missing observations. Section 4 provides a numerical analysis that highlights the computational gain of the matrix approach. Section 5 concludes.

2 An efficient matrix approach

In this section we first recall the standard recursive approach, we then introduce the matrix representation, and finally we show how to render the matrix approach computationally efficient.

2.1 The standard approach to state space models

Consider the general linear Gaussian state space model:

$$\begin{array}{rclll} y_t & = & Z_t\alpha_t + \varepsilon_t, & \varepsilon_t \sim \mathcal{N}(0, H_t), & t = 1, ..., n, \\ \alpha_{t+1} & = & T_t\alpha_t + \eta_t, & \eta_t \sim \mathcal{N}(0, Q_t), & \alpha_1 \sim \mathcal{N}(a_1, P_1). \end{array} \tag{1}$$

The first equation is the *measurement equation* linking the $N \times 1$ vector of observables y_t to the $m \times 1$ state vector α_t. The second equation is the *transition equation* describing the dynamics of the state vector, ε_t and η_t are Gaussian random shocks, and the initial state vector α_1 is also Gaussian distributed. It is usually assumed that $\mathrm{E}(\alpha_1\eta_t') = 0$, $\mathrm{E}(\alpha_1\varepsilon_t') = 0$, and $\mathrm{E}(\varepsilon_t\eta_t') = 0$ $\forall t$, this last assumption can be relaxed at the cost of a slightly complication of the filtering formulae (see Harvey, 1989, sec. 3.2.4).

The *system matrices* Z_t, H_t, T_t, and Q_t are assumed to be non-stochastic[1], as such the observations and the state vector are conditionally Gaussian: $y_t|Y_{t-1} \sim \mathcal{N}(Z_t a_t, F_t)$ and $\alpha_t|Y_{t-1} \sim \mathcal{N}(a_t, P_t)$, where $Y_{t-1} = \{y_{t-1}, ..., y_1\}$ represents the information set at time $t-1$. Thus, the log-likelihood function for the observations, $y = (y_1', \ldots, y_n')'$, can be expressed by the *prediction error decomposition*:

$$\log p(y) = \sum_{t=1}^{n} \log p(y_t|Y_{t-1}) = -\frac{nN}{2}\log 2\pi - \frac{1}{2}\sum_{t=1}^{n}\left(\log|F_t| + v_t'F_t^{-1}v_t\right), \tag{2}$$

where v_t and F_t are recursively computed by the KF:

$$\begin{array}{rclrcll} v_t & = & y_t - Z_t a_t, & F_t & = & Z_t P_t Z_t' + H_t, & \\ K_t & = & T_t P_t Z_t' F_t^{-1}, & L_t & = & T_t - K_t Z_t, & \\ a_{t+1} & = & T_t a_t + K_t v_t, & P_{t+1} & = & T_t P_t L_t' + Q_t, & t = 1, \ldots, n. \end{array} \tag{3}$$

Specifically, $a_t = \mathrm{E}(\alpha_t|Y_{t-1})$ is the *predictive filter* with $P_t = \mathrm{E}[(a_t - \alpha_t)(a_t - \alpha_t)']$ being the mean square error (MSE) matrix. In the case that a proper distribution for α_1 is not available, the filter is initialized with the *diffuse* initial condition (see Harvey, 1989, sec. 3.3.4 and Durbin and Koopman, 2012 ch. 5).

Conditional on the full information set $Y_n = \{y_n, ..., y_1\}$, we have that $\alpha_t|Y_n \sim \mathcal{N}(a_{t|n}, P_{t|n})$, where the conditional moments $a_{t|n} = \mathrm{E}(\alpha_t|Y_n)$ and $P_{t|n} = \mathrm{E}[(a_{t|n} - \alpha_t)(a_{t|n} - \alpha_t)']$ are recursively obtained by the KS:

$$\begin{array}{rclrcll} r_{t-1} & = & Z_t'F_t^{-1}v_t + L_t'r_t, & N_{t-1} & = & Z_t'F_t^{-1}Z_t + L_t'N_tL_t, & \\ a_{t|n} & = & a_t + P_t r_{t-1}, & P_{t|n} & = & P_t - P_t N_{t-1} P_t, & t = n, \ldots, 1, \end{array} \tag{4}$$

with $r_n = 0$ and $N_n = 0$. For more details see Harvey (1989, sec. 3.6) and Durbin and Koopman (2012, sec. 4.4).

[1]Some or all the matrices depend on the unknown parameter vector θ. In order to save on notation we avoid making such reference.

2.2 The matrix representation

Following Durbin and Koopman (2012, sec. 4.13), we express the model (1) as follows:

$$\begin{array}{rcll} y & = & B\alpha + \varepsilon, & \varepsilon \sim \mathcal{N}(0, U), \\ \alpha & = & A(\alpha^* + R\eta), & \eta \sim \mathcal{N}(0, V), \quad \alpha^* \sim \mathcal{N}(a^*, P^*). \end{array} \tag{5}$$

The elements of the measurement equation are:

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, B = \begin{bmatrix} Z_1 & & \\ & \ddots & \\ & & Z_n \end{bmatrix}, \alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}, U = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_n \end{bmatrix}. \tag{6}$$

The elements of the transition equation are:

$$A = \begin{bmatrix} I & & & & \\ T_1 & I & & & \\ T_2T_1 & T_2 & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ T_{n-1}\dots T_1 & T_{n-2}\dots T_2 & \cdots & T_{n-1} & I \end{bmatrix}, \alpha^* = \begin{pmatrix} \alpha_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, R = \begin{bmatrix} 0 & \cdots & 0 \\ I & & \\ & \ddots & \\ & & I \end{bmatrix}, \eta = \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_{n-1} \end{pmatrix},$$

$$V = \begin{bmatrix} Q_1 & & \\ & \ddots & \\ & & Q_{n-1} \end{bmatrix}, a^* = \begin{pmatrix} a_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, P^* = \begin{bmatrix} P_1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}, G = \begin{bmatrix} P_1 & & & \\ & Q_1 & & \\ & & \ddots & \\ & & & Q_{n-1} \end{bmatrix}, \tag{7}$$

where $G = \text{Var}(\alpha^* + R\eta) = (P^* + RVR')$. Let recall the dimensions of the vectors and matrices in the above representation: y and ε are $Nn \times 1$; α, α^* and a^* are $mn \times 1$; η is $m(n-1) \times 1$; B is $Nn \times mn$; U is $Nn \times Nn$; V is $m(n-1) \times m(n-1)$; R is $mn \times m(n-1)$; while A, P^*, and G are $mn \times mn$. It is important to stress that A is block lower triangular matrix, while B, U, and G are block diagonal matrices.

The join distribution of α and y is:

$$\begin{pmatrix} \alpha \\ y \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_\alpha \\ \mu_y \end{bmatrix}; \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha y} \\ \Sigma'_{\alpha y} & \Sigma_{yy} \end{bmatrix} \right), \quad \begin{array}{l} \mu_\alpha = Aa^*, \quad \Sigma_{\alpha\alpha} = AGA', \quad \Sigma_{\alpha y} = \Sigma_{\alpha\alpha}B', \\ \mu_y = B\mu_\alpha, \quad \Sigma_{yy} = B\Sigma_{\alpha\alpha}B' + U. \end{array} \tag{8}$$

The log-likelihood in (2) can then be expressed in the following matrix formulation:

$$\log p(y) = -\frac{nN}{2}\log 2\pi - \frac{1}{2}\left[\log|\Sigma_{yy}| + (y-\mu_y)'\Sigma_{yy}^{-1}(y-\mu_y)\right]. \tag{9}$$

Using the Lemma of the Multivariate Normal, the smoother in (4) can be retrieved as follows:

$$\begin{array}{rcl} \alpha|y & \sim & \mathcal{N}(\mu_{\alpha|y}, \Sigma_{\alpha\alpha|y}) \\ \mu_{\alpha|y} & = & \mu_\alpha + \Sigma_{\alpha\alpha}B'(B\Sigma_{\alpha\alpha}B' + U)^{-1}(y - \mu_y), \\ \Sigma_{\alpha\alpha|y} & = & \Sigma_{\alpha\alpha} - \Sigma_{\alpha\alpha}B'(B\Sigma_{\alpha\alpha}B' + U)^{-1}B\Sigma_{\alpha\alpha}, \end{array} \tag{10}$$

where

$$\mu_{\alpha|y} = \begin{pmatrix} a_{1|n} \\ \vdots \\ a_{n|n} \end{pmatrix}, \quad \Sigma_{\alpha\alpha|y} = \begin{bmatrix} P_{1|n} & \cdots & P_{(1,n)|n} \\ \vdots & \ddots & \vdots \\ P_{(n,1)|n} & & P_{n|n} \end{bmatrix}, \tag{11}$$

the off-diagonal elements of $\Sigma_{\alpha\alpha|y}$ are the cross-covariances $P_{(i,j)|n} = \mathrm{E}[(a_{i|n} - \alpha_i)(a_{j|n} - \alpha_j)']$.

It is possible in principle to compute the log-likelihood and the smoothed state vector without the need of the KF and KS. Unfortunately, the expressions in (9) and (10) involve operations among large matrices making it computationally inefficient, as pointed out by Durbin and Koopman (2012, p.118). This is the reason why the recursive approach has typically been favored in practice.

2.3 Feasible matrix approach

We now show how to compute efficiently both the log-likelihood and the smoother by exploiting operations between vectors and sparse matrices.

Assumption 1 (Invertibility of the System Matrices)**.** *The inverse of G and U exist.*

Given the representation (5)-(7), Assumption 1 implies that the inverse of H_t, Q_t and P_1 exist. By an appropriate specification of the matrix representation this condition is satisfied for a wide range of models. In Appendix B we present few illustrative examples.

Let express the matrix representation (5) as follows:

$$\begin{aligned} y &= B\alpha + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, U), \\ D\alpha &= \alpha^* + R\eta, \quad \eta \sim \mathcal{N}(0, V), \quad \alpha^* \sim \mathcal{N}(a^*, P^*), \end{aligned} \tag{12}$$

where $D = A^{-1}$ is a banded sparse matrix:

$$D = \begin{bmatrix} I & & & \\ -T_1 & I & & \\ & \ddots & \ddots & \\ & & -T_{n-1} & I \end{bmatrix}. \tag{13}$$

Using the Woodbury matrix identity and the results in (8) and (10), the precision matrices $\Sigma_{\alpha\alpha}^{-1} = \Omega_{\alpha\alpha}$ and $\Sigma_{\alpha\alpha|y}^{-1} = \Omega_{\alpha\alpha|y}$ are also banded sparse matrices. Specifically,

$$\Omega_{\alpha\alpha} = D'G^{-1}D = \begin{bmatrix} M_1 & -C_1' & & & \\ -C_1 & M_2 & -C_2' & & \\ & -C_2 & \ddots & \ddots & \\ & & \ddots & M_{n-1} & -C_{n-1}' \\ & & & -C_{n-1} & Q_{n-1}^{-1} \end{bmatrix}, \tag{14}$$

where $C_t = Q_t^{-1}T_t$, $M_t = Q_{t-1}^{-1} + T_t'Q_t^{-1}T_t$, $M_1 = P_1^{-1} + T_1'Q_1^{-1}T_1$, and

$$\Omega_{\alpha\alpha|y} = (\Omega_{\alpha\alpha} + B'U^{-1}B) = \begin{bmatrix} J_1 & -C_1' & & & \\ -C_1 & J_2 & -C_2' & & \\ & -C_2 & \ddots & \ddots & \\ & & \ddots & J_{n-1} & -C_{n-1}' \\ & & & -C_{n-1} & J_n \end{bmatrix}, \tag{15}$$

where $J_t = M_t + Z_t'H_t^{-1}Z_t$ and $J_n = Q_{n-1}^{-1} + Z_n'H_n^{-1}Z_n$.

For diffuse initial condition we delete first m rows and m columns from matrix G and the first m rows from matrix D, as such $\Omega_{\alpha\alpha}$ is singular but $\Omega_{\alpha\alpha|y}$ is non-singular.

Computing the log-likelihood. The quadratic term in (9) is obtained by the following operations among sparse matrices and vectors:[2]

$$(y - \mu_y)'\Sigma_{yy}^{-1}(y - \mu_y) = v'\zeta - \xi'\omega, \tag{16}$$

where

$$v = [y - B(D^{-1}a^*)], \quad \zeta = U^{-1}v, \quad \xi = B'\zeta, \quad \omega = \Omega_{\alpha\alpha|y}^{-1}\xi, \tag{17}$$

the determinant of covariance matrix in (9) is

$$\log|\Sigma_{yy}| = \log|\Omega_{\alpha\alpha|y}| + \log|P_1| + \sum_{t=1}^{n-1}\log|Q_t| + \sum_{t=1}^{n}\log|H_t|, \tag{18}$$

where $\Omega_{\alpha\alpha|y}$ is sparse, while P_1, Q_t and H_t are matrices of small dimension with respect to the overall size of the system. See Appendix A.1 for details.

Computing the smoother. The conditional mean of the smoothed states in (10) can also be retrieved as:

$$\mu_{\alpha|y} = \Omega_{\alpha\alpha|y}^{-1}[\tilde{a}^* + B'(U^{-1}y)], \tag{19}$$

where $\tilde{a}^* = (\tilde{a}_1', 0', \ldots, 0')'$ and $\tilde{a}_1 = P_1^{-1}a_1$; see Appendix A.2 for details. The expression (19) can be efficiently computed as it involves only operations among sparse matrices and vectors. Finally, the covariance $\Sigma_{\alpha\alpha|y}$ is computed as the inverse of the sparse matrix in (15).

3 Additional results

3.1 Weighting function

It is well known that both filtering and smoothing estimators can be expressed as a weighted average of the observations; see Koopman and Harvey (2003). Given the expression (19) and the definition of

[2]Given a $k \times k$ non-singular sparse matrix $\mathcal{S}$ and a $k \times 1$ vector x, we have that $\mathcal{S}^{-1}x = \mathcal{S}\backslash x$, which denotes the unique solution for z to the system $\mathcal{S}z = x$. Following Chan and Jeliazkov (2009), we compute the Cholesky factor C, such that $CC' = \mathcal{S}$, and then we compute $\mathcal{S}^{-1}x = \mathcal{S}\backslash x = C'\backslash(C\backslash x)$, so we have to solve two triangular systems by forward substitution followed by back substitution. Therefore, in total we have three operations and each one requires $\mathcal{O}(k)$ complexity. Finally, we compute $\log\det(\mathcal{S}) = 2\sum_{i=1}^{k}\log c_{ii}$, where c_{ii} are the diagonal elements of C.

B, U, y, $\tilde{a}^*$, we can express the smoothed estimator at time t as follows:

$$a_{t|n} = P_{(t,1)|n}\tilde{a}_1 + \sum_{j=1}^{n} \omega_{t,j} y_j, \quad \text{with} \quad \omega_{t,j} = P_{(t,j)|n} Z_j' H_j^{-1}, \tag{20}$$

where $P_{(t,j)|n} = \mathrm{E}[(a_{t|n} - \alpha_t)(a_{j|n} - \alpha_j)']$. For $j = t$ we have the expression for the weight of the current observation y_t, that is $\omega_{t,t} = P_{t|n} Z_t' H_t^{-1}$. For $t = n$ we obtain the set of weights for the real time filter $a_{n|n}$. It is easy to check that our expressions for the weights $\omega_{t,j}$ match exactly those proposed by Koopman and Harvey (2003) and summarized in Durbin and Koopman (2012, pp.105-106).

Equation (20) highlight that the weights are proportional to the cross-covariances among smoothed estimates $P_{(i,j)|n}$. The full sets of cross-covariances $\Psi_t = [P_{(t,1)|n}, \ldots, P_{(t,n)|n}]$ can be computed efficiently by solving the system of equations $\Psi_t \Omega_{\alpha\alpha|y} = \Upsilon_t$, where $\Upsilon_t = [0_m, \ldots, 0_m, I_m, 0_m, \ldots, 0_m]$ is a selection matrix with identity matrix in the $t - th$ position.

3.2 Missing observations and mixed frequency

One of the advantages of working within a state space framework is that the KF/KS can easily deal with data irregularities, such as missed observations and data sampled at different frequencies. In this section we show how the matrix approach is amended to deal with data irregularities.

Let y_t contain missing data, we define the selection matrix W_t of dimension $N_t \times N$ with $1 \leq N_t \leq N$, such selection matrix eliminates the $i - th$ row from I_N when the $i - th$ variable is missing. Thus, we have that $\tilde{y}_t = W_t y_t$ is the vector of observed variables at time t. The likelihood of the model and the associated smoother can be retrieved applying equations (16)-(19) to the available information. Specifically, the measurement equation of the model becomes $\tilde{y}_t = \tilde{Z}_t \alpha_t + \tilde{\varepsilon}_t$, where $\tilde{Z}_t = W_t Z_t$, $\tilde{\varepsilon}_t = W_t \varepsilon_t$, $\tilde{\varepsilon}_t \sim \mathcal{N}(0, \tilde{H}_t)$, and $\tilde{H}_t = W_t H_t W_t'$. In case no observations are available at time t, we set $W_t = 0_{N \times N}$ such that $\tilde{y}_t$, $\tilde{Z}_t$, and $\tilde{H}_t$ are vector and matrices of zeros.

The case of mixed frequencies is of particular interest for a number of applications, like for instance forecasting low frequency variables using higher frequency predictors (nowcasting). Mixed frequencies typically involve missing observations and temporal aggregation.[3] Specifically, the low frequency indicators can be modeled as a process that is observed at regular low frequency intervals and missing at higher frequency dates, as such this can be easily handled using the matrix approach with the appropriate amendments of the system matrices discussed above.

4 Computational efficiency analysis

In this section we compare the efficiency of the matrix approach with that of the standard recursive approach. Specifically, we report two exercises. First we look at a generic state space model with constant matrices Z, T, H and Q. Second, we look at the VARMA model. Details on the matrix representation of both models are highlighted in Appendix B.

Time-invariant state space model We use the generic state space model with constant system matrices to assess the efficiency of the matrix approach for different dimensions of the model. Specif-

[3]The temporal aggregation requires a modification of the state space representation. The relation between the observed low frequency variable and the corresponding (latent) higher frequency indicator depends on whether the variable is a flow or a stock and on how the variable is transformed before entering the model (see e.g. Banbura et al., 2013).

Table 1: Relative Performance of the Matrix Approach

		(a) Likelihood			(b) Smoother			(c) Smoother & Lik.		
		m			m			m		
n	N	1	5	10	1	5	10	1	5	10
100	1	0.500	0.265	0.510	0.622	0.632	1.690	0.720	0.704	1.835
	5	0.084	0.131	0.253	0.126	0.351	0.943	0.124	0.398	1.031
	10	0.075	0.119	0.233	0.093	0.312	0.867	0.110	0.347	0.941
	30	0.047	0.082	0.147	0.054	0.187	0.495	0.065	0.217	0.568
	100	0.020	0.026	0.045	0.012	0.033	0.098	0.020	0.052	0.111
	200	0.014	0.019	0.036	0.006	0.017	0.036	0.012	0.024	0.055
200	1	0.299	0.202	0.443	0.400	0.574	1.643	0.473	0.626	1.773
	5	0.050	0.097	0.221	0.067	0.313	0.906	0.077	0.340	0.987
	10	0.052	0.091	0.205	0.057	0.284	0.837	0.067	0.310	0.914
	30	0.033	0.066	0.134	0.034	0.169	0.488	0.044	0.193	0.396
	100	0.012	0.020	0.050	0.008	0.029	0.100	0.012	0.038	0.105
	200	0.008	0.018	0.028	0.004	0.016	0.038	0.007	0.020	0.043
500	1	0.179	0.164	0.421	0.266	0.595	1.807	0.312	0.634	1.927
	5	0.031	0.084	0.209	0.042	0.317	1.001	0.049	0.342	1.053
	10	0.031	0.080	0.201	0.038	0.288	0.914	0.046	0.312	0.971
	30	0.022	0.041	0.111	0.023	0.178	0.548	0.030	0.171	0.506
	100	0.008	0.022	0.042	0.006	0.035	0.096	0.007	0.038	0.116
	200	0.005	0.015	0.024	0.003	0.015	0.038	0.004	0.017	0.044
1000	1	0.133	0.156	0.436	0.228	0.620	2.223	0.264	0.658	2.305
	5	0.026	0.080	0.222	0.036	0.341	1.239	0.043	0.361	1.275
	10	0.024	0.073	0.205	0.032	0.300	1.115	0.039	0.320	1.169
	30	0.018	0.057	0.153	0.020	0.190	0.627	0.025	0.191	0.694
	100	0.006	0.020	0.040	0.005	0.037	0.109	0.007	0.038	0.121
	200	0.007	0.014	0.026	0.004	0.016	0.045	0.005	0.018	0.049
2000	1	0.113	0.152	0.632	0.207	0.629	2.454	0.241	0.676	2.621
	5	0.021	0.078	0.318	0.032	0.345	1.341	0.038	0.370	1.436
	10	0.022	0.064	0.313	0.028	0.307	1.246	0.034	0.308	1.353
	30	0.015	0.060	0.200	0.018	0.193	0.729	0.022	0.197	0.783
	100	0.009	0.022	0.050	0.006	0.035	0.125	0.008	0.040	0.139
	200	0.006	0.014	0.028	0.004	0.016	0.050	0.005	0.018	0.055

Notes: We simulate the model 101 times (on Matlab R2017a, with Intel Core i7-7700K and 4.20 GHz CPU) and take the median value of the computational time for each of the two methods. The table reports the ratio of the computational time of the matrix approach over the recursive approach. Values below one (in grey) highlight that the matrix approach is more efficient. The dimensions of the state space model (1) are: $m = \dim(\alpha_t)$, $N = \dim(y_t)$, and n is the sample size. The corresponding dimension for the matrix representation (5) is $\dim(\alpha) = mn$.

ically, Table 1 reports the relative performance of the matrix approach compared with the traditional recursive approach with different N and n, i.e. the cross-section and time series dimension of the vector of observables, and m, the length of the state vector. We then look at three possible scenarios: one where we only compute the likelihood of the model (Panel a), one where we only compute the smoother (Panel b) and the combined case where we compute both the likelihood and the smoother (Panel c).

When a model is estimated by ML the likelihood needs to be computed repeatedly and the results reported in Panel (a) are the ones of interest. This case is also of interest if one is using a Metropolis

Table 2: Relative Performance for VARMA(1,1)

	Likelihood				
N\n	100	200	500	1000	2000
1	0.126	0.069	0.043	0.037	0.028
3	0.089	0.066	0.047	0.044	0.038
5	0.103	0.076	0.066	0.063	0.055
7	0.115	0.091	0.079	0.073	0.069
10	0.159	0.134	0.118	0.102	0.106
15	0.121	0.115	0.103	0.119	0.120

Notes: For each DGP we simulate the models 101 times and take the median computational time, the table reports the ratio of the matrix approach over the standard recursive KF/KS. For the computations we use Matlab R2017a (on an Intel Core i7-7700K and 4.20 GHz CPU).

step within a Gibbs sampler (see e.g. Chib and Greenberg, 1995, and Geweke and Tanizaki, 2001) and for MCMC methods for classical estimation (Chernozhukov and Hong, 2003), as the rejection step in these cases requires the evaluation of the likelihood for each of the proposal draws.[4] Panel (b) and (c) are of interest if the model is estimated using the EM algorithm. In this case one needs to compute the vector of the smoothed stated and the associated covariance matrix in order to update the estimates of the coefficient of the model, and the likelihood is required in order to devise a stopping rule for the algorithm (see Shumway and Stoffer, 1982, and Banbura and Modugno, 2014).

The results in Table 1 highlight how the matrix approach is a competitive alternative to the standard recursive approach. In fact, for most of the cases considered the ratio of computational time is below 1, indicating the matrix approach is more efficient.[5] The gains are particularly accentuated for models featuring large datasets (i.e. large N and n).

The matrix approach is always more efficient for the computation of the likelihood, whereas when it comes to computing the smoother it becomes inefficient for $m >> N$, and this is due to the inverse of $\Omega_{\alpha\alpha|y}$ that has dimension equal to $\dim(\alpha)$. It is also worth noting that, while in Table 1 we have assumed that $\dim(\alpha) = mn$, in practice it is often possible to find a convenient representation of the model that allows to reduce $\dim(\alpha)$, therefore making the matrix approch more efficient.[6] For instance, take the case of $N = 1$ and $m = 10$, a realistic setting which would give rise to a model with these dimensions is a univariate trend-cycle-seasonal model. This model can easily be re-written in a matrix representation so that $\dim(\alpha) = n$ rather then mn; see Appendix B.3 for details. The possibility of a matrix representation that reduces the dimensionality of the problem arises also for VARMA models as we highlight in the next example.

VARMA models As a second exercise we look at the relative performance of the matrix approach in computing the likelihood for a VARMA model. Specifically, we focus on VARMA(1,1) models of increasing dimensions. This case is of interest because the state space representation leads to a zero measurement error and $m = 2N$. A point worth highlighting here is that the matrix form can be accommodated so that $\dim(\alpha) = \dim(y) = Nn$ rather than $2Nn$; see Appendix B.4 for details.

[4] This case is also of interest for the estimation of DSGE models (see e.g. An and Schorfheide, 2007).

[5] In Appendix C we report the relative performance of the fast state smoother (Durbin and Koopman, 2012, sec. 4.6.2). In this case we do not compute covariance matrix of $\Sigma_{\alpha\alpha|y}$, and the matrix approach is always more efficient.

[6] See, e.g., Appendix B.

Table 2 highlights how the matrix approach is always more efficient than the recursive approach, with a gain in computational times ranging from 86% to 97%.

5 Conclusion

In this paper we propose an efficient matrix approach for estimating state space models in the classical framework without using the KF and KS. We highlight how the matrix approach is not only tractable but often computationally more efficient than the traditional recursive approach. This is particularly true for 'large data' settings, i.e. situations where the number of observable variables and their time series dimension is large. Moreover, we also derive expressions for the weighting function and highlight how to deal with missing data.

References

An, S. and F. Schorfheide (2007). Bayesian Analysis of DSGE Models. *Econometric Reviews*, 26(2-4), 113-172.

Banbura M., Giannone D., Modugno M. and L. Reichlin (2013). Now-Casting and the Real-Time Data Flow. *Handbook of Economic Forecasting*, Elsevier.

Banbura M. and M. Modugno (2014). Maximum Likelihood Estimation of Factor Models on Datasets with Arbitrary Pattern of Missing Data, *Journal of Applied Econometrics*, 29(1), 133-160.

Chan J.J. and I. Jeliazkov (2009). Efficient Simulation and Integrated Likelihood Estimation in State Space Models. *International Journal of Mathematical Modelling & Numerical Optimisation*, 1, 101-120.

Chernozhukov V. and H. Hong (2003). An MCMC approach to classical estimation, *Journal of Econometrics*, 115(2), 293-346.

Chib S. and E. Greenberg (1995). Understanding the Metropolis Hastings algorithm. *American Statistician*, 49, 327-335.

Durbin J. and SJ. Koopman (2012). *Time Series Analysis by State Space Methods.* Oxford University Press.

Geweke J. and H. Tanizaki (2001). Bayesian estimation of state-space models using the Metropolis-Hastings algorithm within Gibbs sampling. *Computational Statistics & Data Analysis*, 37(2), 151-170.

Giannone D., Reichlin L. and D. Small (2008). Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, 55(4), 665-676.

Harvey A.C. (1989). *Forecasting, structural time series models and Kalman filter.* Cambridge University Press.

Harvey A.C. and R. Pierce (1984). Estimating Missing Observations in Economic Time Series. *Journal of the American Statistical Association*, 79, 125-131.

Koopman SJ. and A.C. Harvey (2003). Computing observation weights for signal extraction and filtering. *Journal of Economic Dynamics & Control*, 27, 1317-1333.

Lütkepohl, H. (2007). *New Introduction to Multiple Time Series Analysis.* Springer.

Mariano R.S. and Y. Murasawa (2003). A new coincident index of business cycles based on monthly and quarterly series. *Journal of Applied Econometrics*, 18(4), 427-443.

McCausland W.J., Miller S. and D. Pelletier (2011). Simulation smoothing for state-space models: A computational efficiency analysis. *Computational Statistics & Data Analysis*, 55, 199-212.

Shumway, R. and D. Stoffer (1982). An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3, 253-264.

Stock J.H. and M.W. Watson (2010). Dynamic Factor Models. In *Oxford Handbook of Economic Forecasting*, Michael P. Clements and David F. Hendry (eds), Oxford University Press.

A Derivations

A.1 The log-likelihood

Using the Woodbury matrix identity we have that

$$\Sigma_{yy}^{-1} = (B\Sigma_{\alpha\alpha}B' + U)^{-1} = U^{-1} - U^{-1}B(\Sigma_{\alpha\alpha}^{-1} + B'U^{-1}B)^{-1}B'U^{-1} = U^{-1} - U^{-1}B\Sigma_{\alpha\alpha|y}B'U^{-1}.$$

Using the matrix determinant Lemma and given that A is block lower triangular we have that

$$\det(\Sigma_{yy}) = \det(B\Sigma_{\alpha\alpha}B' + U) = \det(\Sigma_{\alpha\alpha}^{-1} + B'U^{-1}B)\det(\Sigma_{\alpha\alpha})\det(U) = \det(\Sigma_{\alpha\alpha|y}^{-1})\det(G)\det(U).$$

A.2 The smoother

Rearranging the conditional mean in (10) we have:

$$\mu_{\alpha|y} = [I - \Sigma_{\alpha\alpha}B'(B\Sigma_{\alpha\alpha}B' + U)^{-1}B]\mu_\alpha + \Sigma_{\alpha\alpha}B'(B\Sigma_{\alpha\alpha}B' + U)^{-1}y.$$

Using other rules of the matrix inversion Lemma we that:

$$\Sigma_{\alpha\alpha}B'(B\Sigma_{\alpha\alpha}B' + U)^{-1} = (\Sigma_{\alpha\alpha}^{-1} + B'U^{-1}B)^{-1}B'U^{-1} = \Sigma_{\alpha\alpha|y}B'U^{-1},$$

$$[I - \Sigma_{\alpha\alpha}B'(B\Sigma_{\alpha\alpha}B' + U)^{-1}B] = (I + \Sigma_{\alpha\alpha}B'U^{-1}B)^{-1} = (\Sigma_{\alpha\alpha}^{-1} + B'U^{-1}B)^{-1}\Sigma_{\alpha\alpha}^{-1} = \Sigma_{\alpha\alpha|y}\Sigma_{\alpha\alpha}^{-1}.$$

Moreover, it turns out that

$$\Sigma_{\alpha\alpha}^{-1}\mu_\alpha = (D'G^{-1}D)Aa^* = D'G^{-1}a^* = (\tilde{a}_1', 0', \ldots, 0')' = \tilde{a}^*,$$

with $\tilde{a}_1 = P_1^{-1}a_1$. Putting all together we obtain

$$\mu_{\alpha|y} = \Sigma_{\alpha\alpha|y}(\tilde{a}^* + B'U^{-1}y).$$

B Examples (For Online Publication)

In this Appendix we first specialize the description in Section 2 for a time invariant state space model, then we show how to (efficiently) cast some popular models in the matrix form.

B.1 Time invariant state space model

The matrix representation of a state space model (1) with constant system matrices is

$$\begin{array}{rcll} y & = & B\alpha + \varepsilon, & \varepsilon \sim \mathcal{N}(0, U), \\ D\alpha & = & \alpha^* + R\eta, & \eta \sim \mathcal{N}(0, V), \quad \alpha^* \sim \mathcal{N}(a^*, P^*), \end{array} \tag{B.1}$$

where y, α, ε, α^*, R, η, a^*, and P^* are the same as in (6)-(7), while

$$D = \begin{bmatrix} I & & & \\ -T & I & & \\ & \ddots & \ddots & \\ & & -T & I \end{bmatrix}, \quad B = (I_n \otimes Z), \quad U = (I_n \otimes H), \quad V = (I_{n-1} \otimes Q). \tag{B.2}$$

Given invertible covariance matrices H, Q and P_1 the resulting banded sparse precision matrix is

$$\Omega_{\alpha\alpha|y} = \begin{bmatrix} J_1 & -C' & & & \\ -C & J & -C' & & \\ & -C & \ddots & \ddots & \\ & & \ddots & J & -C' \\ & & & -C & J_n \end{bmatrix}, \tag{B.3}$$

where $C = Q^{-1}T$, $J = Q^{-1} + T'Q^{-1}T + Z'H^{-1}Z$, $J_1 = P_1^{-1} + T'Q^{-1}T + Z'H^{-1}Z$, and $J_n = Q^{-1} + Z'H^{-1}Z$. The log-likelihood function and the smoothed state vector are efficiently computed as in Section 2.3. In case of singularities in H and Q, the matrix representation can be accomodated to have well defined precision matrices as it is shown in the examples below.

B.2 Factor model

Consider that the $N \times 1$ vector y_t follows the dynamic factor model (Stock and Watson, 2010):

$$\begin{array}{rll} y_t & = \Lambda f_t + \varepsilon_t, & \varepsilon_t \sim \mathcal{N}(0, \Sigma_\epsilon), \qquad t = 1, ..., n, \\ f_{t+1} & = \Phi_1 f_t + \Phi_2 f_{t-1} + \eta_t, & \eta_t \sim \mathcal{N}(0, \Sigma_\eta), \end{array} \tag{B.4}$$

where f_t is the $r \times 1$ vector of unobserved factors, ε_t and η_t are random shock of dimension $N \times 1$ and $r \times 1$ respectively, Λ, Φ_1, Φ_2, Σ_ϵ, and Σ_η are matrices of appropriate dimension. The standard state space representation of model (B.4) leads to $m = 2r$ and the covariance matrix of the transition equation is singular. The matrix representation can be appropiately accomodated such that $\dim(\alpha) = rn$ and the covariance matrices are non-singular. Specifically,

$$\begin{array}{rcll} y & = & B\alpha + \varepsilon, & \varepsilon \sim \mathcal{N}(0, I_n \otimes \Sigma_\epsilon), \\ D\alpha & = & \alpha^* + R\eta, & \eta \sim \mathcal{N}(0, I_{n-1} \otimes \Sigma_\eta), \quad \alpha^* \sim \mathcal{N}(a^*, P^*), \end{array} \tag{B.5}$$

where $y = (y_1', \ldots, y_n')'$, $B = diag[0, (I_n \otimes \Lambda)]$, $\alpha = (f_0', f_1', \ldots, f_n')'$, $\varepsilon = (\varepsilon_1', \ldots, \varepsilon_n')'$, $\alpha^* = (f_0', f_1', 0', \ldots, 0')'$, $\eta = (\eta_1', \ldots, \eta_{n-1}')'$,

$$D = \begin{bmatrix} I & & & & \\ 0 & I & & & \\ -\Phi_2 & -\Phi_1 & I & & \\ & \ddots & \ddots & \ddots & \\ & & -\Phi_2 & -\Phi_1 & I \end{bmatrix}, \quad R = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ I & & \\ & \ddots & \\ & & I \end{bmatrix}, \quad G = \begin{bmatrix} P_1 & & & & \\ & \Sigma_\eta & & & \\ & & \ddots & & \\ & & & \Sigma_\eta & \end{bmatrix},$$

where $G = \mathrm{Var}(\alpha^* + R\eta)$, and P_1 is the unconditional variance of the vector $(f_0', f_1')'$. The precision matrix $\Omega_{\alpha\alpha|y} = [\Omega_{\alpha\alpha} + B'(I_n \otimes \Sigma_\varepsilon^{-1})B]$, where $\Omega_{\alpha\alpha} = D'G^{-1}D$, and $B'(I_n \otimes \Sigma_\varepsilon^{-1})B = diag(0, [I_n \otimes (\Lambda'\Sigma_\varepsilon^{-1}\Lambda)])$. Thus, the smoother moments are efficiently computed as follows:

$$\mu_{\alpha|y} = \Omega_{\alpha\alpha|y}^{-1}(0', \xi')', \quad \xi = (I_n \otimes \Lambda'\Sigma_\epsilon^{-1})y, \quad \Sigma_{\alpha\alpha|y} = \Omega_{\alpha\alpha|y}^{-1}. \tag{B.6}$$

Note that $a^* = 0$, thus $\log p(y) = -\frac{nN}{2}\log 2\pi - \frac{1}{2}(\log|\Sigma_{yy}| - y'\Sigma_{yy}^{-1}y)$ can be efficiently computed as:

$$\begin{gathered} \log|\Sigma_{yy}| = \log|\Omega_{\alpha\alpha|y}| + \log|P_1| + (n-1)\log|\Sigma_\eta| + n\log|\Sigma_\epsilon|, \\ y'\Sigma_{yy}^{-1}y = y'\zeta - (0', \xi')\mu_{\alpha|y}, \quad \zeta = (I_n \otimes \Sigma_\epsilon^{-1})y. \end{gathered} \tag{B.7}$$

B.3 Unobserved components model

Let assume that the univariate variable y_t follows the trend-cycle model:

$$\begin{aligned} y_t &= \tau_t + c_t, && t = 1, \ldots n, \\ \tau_{t+1} &= \tau_t + \beta_t, && \\ \beta_{t+1} &= \beta_t + u_{\tau t}, && u_{\tau t} \sim \mathcal{N}(0, \sigma_\tau^2), \\ c_{t+1} &= \phi_1 c_t + \phi_2 c_{t-1} + u_{ct}, && u_{ct} \sim \mathcal{N}(0, \sigma_c^2). \end{aligned} \tag{B.8}$$

We express (B.8) in the following matrix form:

$$\begin{aligned} y &= \tau + c, && \\ D_\tau \tau &= u_\tau, && u_\tau \sim \mathcal{N}(0, \sigma_\tau^2 I_{n-2}), \\ D_c c &= c^* + R u_c, && u_c \sim \mathcal{N}(0, \sigma_c^2 I_{n-2}), \quad c^* \sim \mathcal{N}(\mu_{c^*}, \Sigma_{c^*c^*}), \end{aligned} \tag{B.9}$$

where $y = (y_1, \ldots, y_n)'$, $\tau = (\tau_1, \ldots, \tau_n)'$, $c = (c_1, \ldots, c_n)'$, $u_\tau = (u_{\tau 1}, \ldots, u_{\tau n-2})'$, $u_c = (u_{c2}, \ldots, u_{cn-1})'$, $c^* = (c_1, c_2, 0, \ldots, 0)'$,

$$D_\tau = \begin{bmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{bmatrix}, \quad D_c = \begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ -\phi_2 & -\phi_1 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -\phi_2 & -\phi_1 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix},$$

μ_{c^*} and $\Sigma_{c^*c^*}$ are the unconditional moments of c^*. Because of the diffuse initial condition D_τ is rank deficient.[1] It is worth to note that the state space representation of model (B.8) leads to $m = 4$, the matrix representation (B.9) leads to matrices and vectors of dimension n rather then mn. This is generally true for UC models with multiple components (e.g. trend, cycle, seasonal) for which we can express the matrix form as the sum of vectors of length n.

The distribution for the two components and the observations reads as:

$$\begin{aligned}
&\tau \sim \mathcal{N}(0, \Omega_{\tau\tau}^{-1}), \quad \Omega_{\tau\tau} = D_\tau' G_\tau^{-1} D_\tau, \qquad G_\tau = \sigma_\tau^2 I_{n-2}, \\
&c \sim \mathcal{N}(0, \Omega_{cc}^{-1}), \quad \Omega_{cc} = D_c' G_c^{-1} D_c, \qquad G_c = (\Sigma_{c^*c^*} + \sigma_\tau^2 RR') \\
&y \sim \mathcal{N}(0, \Sigma_{yy}), \quad \Sigma_{yy} = (\Omega_{\tau\tau}^{-1} + \Omega_{cc}^{-1}), \qquad \Omega_{\tau\tau}^{-1} = D_\tau^+ G_\tau D_\tau^{+\prime}, \quad \Omega_{cc}^{-1} = D_c^{-1} G_c D_c^{-1\prime}
\end{aligned}$$

where D_τ^+ is the right inverse of D_τ. The smoother estimators for the two components are:

$$\mu_{\tau|y} = \Omega_{\tau\tau|y}^{-1} \Omega_{cc} y, \qquad \mu_{c|y} = \Omega_{cc|y}^{-1} \Omega_{\tau\tau} y, \qquad \Omega_{\tau\tau|y} = \Omega_{cc|y} = (\Omega_{\tau\tau} + \Omega_{cc})$$

where $\Omega_{\tau\tau}$, Ω_{cc}, $\Omega_{\tau\tau|y}$, $\Omega_{cc|y}$ are all banded sparse matrices. Moreover, we have the following identities:

$$\Sigma_{yy}^{-1} = \Omega_{yy} = \Omega_{\tau\tau} - \Omega_{\tau\tau}\Omega_{\tau\tau|y}^{-1}\Omega_{\tau\tau} = \Omega_{cc} - \Omega_{cc}\Omega_{cc|y}^{-1}\Omega_{cc},$$

and $\log p(y) = -\frac{n}{2}\log 2\pi + \frac{1}{2}\log|\Omega_{yy}| - \frac{1}{2} y'\Omega_{yy} y$ can be computed efficiently noting that:

$$y'\Omega_{yy} y = y'\zeta - \zeta'\xi, \quad \zeta = \Omega_{\tau\tau} y, \quad \xi = \Omega_{\tau\tau|y}^{-1}\zeta.$$

B.4 Vector autoregressive moving average model

Assume that the $N \times 1$ vector of observable variables y_t follows the VARMA(1,1) model:

$$y_t = \Phi y_{t-1} + \varepsilon_t + \Theta\varepsilon_{t-1}, \quad \varepsilon_t \sim \mathcal{N}(0, \Sigma), \quad t = 1, \ldots, n. \tag{B.10}$$

The matrix representation of model (B.10) is:

$$D_\phi y = \alpha^* + D_\theta \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I_n \otimes \Sigma), \quad \alpha^* \sim \mathcal{N}(a^*, P^*), \tag{B.11}$$

where $y = (y_1', \ldots, y_n')'$, $\alpha^* = (y_1', 0', \ldots, 0')'$, $\varepsilon = (\varepsilon_1', \ldots, \varepsilon_n')'$, and

$$D_\phi = \begin{bmatrix} I & & & \\ -\Phi & I & & \\ & \ddots & \ddots & \\ & & -\Phi & I \end{bmatrix}, \quad D_\theta = \begin{bmatrix} 0 & & & \\ \Theta & I & & \\ & \ddots & \ddots & \\ & & \Theta & I \end{bmatrix},$$

It is worth to note that the state space representation of (B.10) usually implies $m = \dim(\alpha_t) = 2N$, while the matrix representation (B.11) leads to have that $\dim(y) = Nn$. Specifically, $y \sim \mathcal{N}(0, \Omega_{yy}^{-1})$,

[1] Alternatively, we can specify a full rank matrix D_τ and, in line with Harvey (1989, sec. 3.3.4), we set a large variance for the initial vector $(\tau_1, \tau_2)'$.

where $\Omega_{yy} = D_\phi' G^{-1} D_\phi$, and $G = [P^* + D_\theta (I_n \otimes \Sigma) D_\theta']$ is an invertible banded sparse matrix:

$$G = \begin{bmatrix} \Gamma_0 & & & & \\ & M & C' & & \\ & C & M & \ddots & \\ & & \ddots & \ddots & C' \\ & & & C & M \end{bmatrix},$$

with Γ_0 being the unconditional variance of y_1, $M = (\Sigma + \Theta\Sigma\Theta')$, and $C = \Theta\Sigma$. Therefore, the log-likelihood can be efficiently computed as follows

$$\log p(y) = -\frac{nN}{2} \log 2\pi - \frac{1}{2} \left(\log |G| + \zeta' \xi \right), \quad \zeta = D_\phi y, \quad \xi = G^{-1} \zeta. \tag{B.12}$$

The representation (B.11) can be also found in Lütkepohl (2007, sec. 12.2.3) although it is typically never used in practice.

C Additional results (For Online Publication)

In this Appendix we report the performance of the matrix approach against the fast state smoother (Durbin and Koopman, 2012, sec. 4.6.2).

Table C.1: Relative Performance for Fast Smoother

		Fast Smoother		
		m		
n	N	1	5	10
100	1	0.180	0.118	0.206
	5	0.066	0.075	0.133
	10	0.050	0.072	0.129
	30	0.038	0.059	0.095
	100	0.016	0.024	0.036
	200	0.011	0.016	0.030
200	1	0.098	0.091	0.176
	5	0.031	0.061	0.114
	10	0.033	0.060	0.110
	30	0.029	0.050	0.100
	100	0.011	0.019	0.041
	200	0.007	0.017	0.026
500	1	0.057	0.074	0.202
	5	0.018	0.046	0.136
	10	0.019	0.048	0.126
	30	0.020	0.045	0.115
	100	0.008	0.022	0.041
	200	0.005	0.014	0.026
1000	1	0.041	0.068	0.205
	5	0.013	0.045	0.135
	10	0.014	0.045	0.134
	30	0.017	0.050	0.112
	100	0.007	0.021	0.039
	200	0.010	0.019	0.030
2000	1	0.033	0.075	0.310
	5	0.010	0.050	0.180
	10	0.013	0.050	0.196
	30	0.015	0.049	0.145
	100	0.010	0.024	0.047
	200	0.012	0.020	0.031

Notes: For each DGP we simulate the models 101 times and take the median computational time, the table reports the ratio of the matrix approach over the standard recursive KF/KS. For the computations we use Matlab R2017a (on an Intel Core i7-7700K and 4.20 GHz CPU).