Numerical integration of the continuous time state space model

Granville Tunnicliffe Wilson*

March 1, 2019

Abstract

Keywords: Continuous time series, irregularly sampled time series, state space model integration, Padé approximation, square root algorithm

1 The continuous time state space model

We write the continuous time stochastic state space model as

$$dS(t) = TS(t)dt + dE(t) \tag{1}$$

where the state vector $S(t)$ is of dimension d and $E(t)$ is a multivariate Wiener process. Our particular application of this model is to the representation of the continuous time vector ZAR (VCZAR) model presented in Tunnicliffe Wilson et al. (2015, chap. 7) for a multivariate time series $x(t)$. In that case the state dimension is $d = p \times m$ where m is the dimension of the series and p the model order. The transition matrix T is constant in time and the incremental error $dE(t)$ has variance matrix Vdt where V is also constant but sparse. This is because $dE(t)$ is zero apart from the first m entries equal to model innovation process $de(t)$ with variance $V_e dt$. Thus V has entries only in the upper left $m \times m$ block set to V_e.

In Tunnicliffe Wilson et al. (2015, chap. 8) this model is fitted to irregularly sampled time series data using state space filtering algorithms, for which purpose it is required to integrate (1) over the time period r between successive sampling times. This integration yields the discrete transition step

$$S(t) = MS(t-r) + w(r) \tag{2}$$

where

$$M = \exp(rT) \tag{3}$$

and $w(r)$ has variance matrix

$$W = \int_0^r \exp(hT)V\exp(hT')dh. \tag{4}$$

*Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YF, UK.

In MATLAB the matrix exponential M in (3) may be computed by a standard function which may also be used to compute the integral in (4) using the result for partitioned matrices

$$\exp\begin{pmatrix} -rT & rV \\ 0 & rT' \end{pmatrix} = \begin{pmatrix} M^{-1} & M^{-1}W \\ 0 & M' \end{pmatrix}. \tag{5}$$

The Appendix presents a derivation of this result, which is standard in the systems literature and was communicated to me, together with the efficient computational procedure described in the remainder of this paper, by Juan-Carlos Ibáñez. From (3) and (5) the variance quantity W in (4) is readily derived and it would appear that the standard MATLAB matrix exponential is sufficient for the state space computations. However, the numerical stability of the filter updating step is much improved by using what is known as the square root algorithm. This requires the variance factor H which satisfies $H'H = W$. The improvements in numerical stability may be sacrificed if H is derived by factorizing W. They are retained by computing H directly from T and the factor G_e of $V_e = G_e'G_e$. The method for doing this is now described. It is derived by application of the Padé approximation of the exponential function.

2 Padé approximation of the exponential function

The exponential function for a square matrix A is defined in the same manner as for a scalar, by the series

$$\exp(A) = \sum_{k=0}^{\infty} A^k/k! \tag{6}$$

which is always convergent. A numerical approximation may be computed by truncating the series. For such an approximation and for the Padé approximation given below, a valuable strategy to improve numerical accuracy is first to compute the approximation for the argument $A_K = A/s$ where $s = 2^K$ is a scaling factor for some choice of integer $K \geq 0$. Let $M_k = \exp(A_k)$. Then the squaring sequence $M_k = M_{k+1}^2$ for $k = K-1 \ldots 0$ gives the approximation M_0 of $\exp(A)$. For the Padé approximation s is chosen so that the norm of A_K is less than 0.4.

The Padé approximation of order q is of the form

$$\exp(A) \approx \frac{\sum_{k=0}^{q} c_{q,k}A^k}{\sum_{k=0}^{q} c_{q,k}(-A)^k} = \frac{N_q(A)}{D_q(A)}, \tag{7}$$

where the numerator and denominator commute, so the ratio can be written as either $N_q(A)D_q(A)^{-1}$ or $D_q(A)^{-1}N_q(A)$. The coefficients in the approximation depend on both the index k and the order q and are given by

$$c_{q,k} = \frac{2q-1)!q!}{(2q)!k!(q-k)!} \tag{8}$$

or recursively by

$$c_{q,k} = 1 \text{ for } k = 0; \quad c_{q,k} = c_{q,k-1}\frac{q-k+1}{k(2q-k+1)} \text{ for } k = 1 \ldots q. \tag{9}$$

For the scalar exponential function the absolute error in the Padé approximation of order $q = 6$ over the unit interval is less than 5×10^{-13}.

We now consider the efficient construction of the Padé approximation to the exponential on the left of (5). For this we require the powers of the partitioned matrix in the exponential argument:

$$\begin{pmatrix} -rT & rV \\ 0 & rT' \end{pmatrix}^k = \begin{pmatrix} (-rT)^k & R_k \\ 0 & (rT')^k \end{pmatrix}. \tag{10}$$

where R_k can be efficiently calculated by the recursion $R_k = -rTR_{k-1} + rV(rT')^{k-1}$. From this, for later use, we can determine by simple induction that

$$R_k = \sum_{i=0}^{k-1} (-rT)^i \, rV(rT')^{k-1-i}. \tag{11}$$

The numerator and denominator of the required Padé approximation are therefore of the form

$$N = \begin{pmatrix} N_{1,1} & N_{1,2} \\ 0 & N_{2,2} \end{pmatrix}, \quad D = \begin{pmatrix} D_{1,1} & D_{1,2} \\ 0 & D_{2,2} \end{pmatrix} \tag{12}$$

where

$$N_{1,1} = \sum_{k=0}^{q} c_{q,k}(-rT)^k, \quad D_{1,1} = \sum_{k=0}^{q} c_{q,k}(rT)^k, \tag{13}$$

$N_{2,2} = D'_{1,1}$, $D_{2,2} = N'_{1,1}$ and

$$N_{1,2} = \sum_{k=0}^{q} c_{q,k} R_k, \quad D_{1,2} = \sum_{k=0}^{q} c_{q,k}(-1)^k R_k. \tag{14}$$

Now set the partitioned Padé approximation to

$$F = \begin{pmatrix} F_{1,1} & F_{1,2} \\ 0 & F_{2,2} \end{pmatrix} = N\,D^{-1} = \begin{pmatrix} N_{1,1} & N_{1,2} \\ 0 & N_{2,2} \end{pmatrix} \begin{pmatrix} D_{1,1} & D_{1,2} \\ 0 & D_{2,2} \end{pmatrix}^{-1} \tag{15}$$

and on equating

$$\begin{pmatrix} F_{1,1} & F_{1,2} \\ 0 & F_{2,2} \end{pmatrix} \begin{pmatrix} D_{1,1} & D_{1,2} \\ 0 & D_{2,2} \end{pmatrix} = \begin{pmatrix} N_{1,1} & N_{1,2} \\ 0 & N_{2,2} \end{pmatrix} \tag{16}$$

we can derive, allowing that $D_{1,1}$ and $N_{1,1}$ commute,

$$F_{1,2} = D_{1,1}^{-1}\,(D_{1,1}N_{1,2} - N_{1,1}D_{1,2})\,D_{2,2}^{-1}. \tag{17}$$

This is the approximation to the term $M^{-1}W$ in (5), the approximation to M being $N_{1,1}^{-1}D_{1,1}$. From their product we have, using also $D_{2,2} = N'_{1,1}$, the approximation

$$W \approx N_{1,1}^{-1}\,(D_{1,1}N_{1,2} - N_{1,1}D_{1,2})\,N'^{-1}_{1,1}. \tag{18}$$

The quantities in this expression are readily calculated from formulae given above, and on factorizing $D_{1,1}N_{1,2} - N_{1,1}D_{1,2} = R'R$ the right factor of W required for the square root filter is given by $H = RN'^{-1}_{1,1}$. It is possible, however, to form the factor R directly from T and the factor G_e of V_e. This improves the numerical conditioning and is described in the next section.

3 Direct computation of the factored variance integral

The method is to expand the quantity $K = D_{1,1}N_{1,2} - N_{1,1}D_{1,2}$ in (18) so that it can be directly factored as $R'R$. Bringing together (11), (13) and (14) we have

$$D_{1,1}N_{1,2} = \left\{ \sum_{\ell=0}^{q} c_{q,\ell}(rT)^{\ell} \right\} \left[\sum_{k=0}^{q} c_{q,k} \left\{ \sum_{i=0}^{k-1} (-rT)^i \, rV \, (rT')^{k-1-i} \right\} \right] \tag{19}$$

and

$$N_{1,1}D_{1,2} = \left\{ \sum_{\ell=0}^{q} c_{q,\ell}(-rT)^{\ell} \right\} \left[\sum_{k=0}^{q} c_{q,k}(-1)^k \left\{ \sum_{i=0}^{k-1} (-rT)^i \, rV \, (rT')^{k-1-i} \right\} \right]. \tag{20}$$

We can now express

$$K = \sum_{s=0}^{2q-1} \sum_{t=0}^{q-1} v_{s,t} T^s V (T')^t \, r^{s+t+1} \tag{21}$$

where, formally,

$$v_{s,t} = \sum_{\ell=0}^{q} \sum_{k=0}^{q} \sum_{i=0}^{k-1} c_{q,\ell} \, c_{q,k} \left\{ (-1)^i - (-1)^{\ell+k+i} \right\} I_{s=\ell+i} \, I_{t=k-1-i} \tag{22}$$

which can be reduced, by setting $i = k - 1 - t$ and $\ell = s - i = s + t + 1 - k$, to

$$v_{s,t} = \sum_{k=1+t}^{\min(s+t+1,q)} c_{q,s+t-k+1} \, c_{q,k} \left\{ (-1)^{k-1-t} - (-1)^{s+k} \right\}. \tag{23}$$

On further examination of this expression some identically canceling terms can be removed. It can also be confirmed that $v_{s,t}$ is symmetric, with common upper limit of $q-1$ on s and t (as already shown in (22) and is zero for odd values of $s+t$. On noting that s and t have the same parity, the symmetry is evident in the final form for $s+t$ even:

$$v_{s,t} = 2 \sum_{k=1+\max s,t}^{\min(s+t+1,q)} c_{q,s+t-k+1} \, c_{q,k} \, (-1)^{t+k+1}. \tag{24}$$

We now explain how K as presented in (21), but with the upper limit on s reduced to $q-1$, can be efficiently factored as $R'R$.

The first step is to form the right factor of the positive definite matrix with elements $v_{s,t}$ so we can express for $t \geq s$

$$v_{s,t} = \sum_{k=0}^{s} w_{k,s} w_{k,t}. \tag{25}$$

Then define the matrix B consisting of vertically stacked block matrices for $k = 0 \ldots q-1$

$$B_k = \sum_{t=k}^{q-1} w_{k,t} G (rT')^t. \tag{26}$$

where G is the right factor of V. For our applications this will be of size $m \times d$ with the upper $m \times m$ elements set to G_e, so that B_k is of size $m \times d$ and B is $qm \times d$. The computations in the construction of B_k are reduced by noting that, as for $v_{r,s}$, the elements $w_{k,t} = 0$ for $k + t$ odd.

We confirm that

$$
\begin{aligned}
r\, B'\, B &= r \sum_{k=0}^{q-1} B_k' B_k && (27)\\
&= r \sum_{k=0}^{q-1} \left\{ \sum_{s=k}^{q-1} (rT)^s G' w_{k,s} \right\} \left\{ \sum_{t=k}^{q-1} w_{k,t} G (rT')^t \right\} && (28)\\
&= r \sum_{s=0}^{q-1} \sum_{t=0}^{q-1} \left(\sum_{k=0}^{\min(s,t)} w_{k,s} w_{k,t} \right) (rT)^s G' G (rT')^t && (29)\\
&= \sum_{s=0}^{q-1} \sum_{t=0}^{q-1} v_{s,t} T^s V (T')^t \, r^{s+t+1} = K. && (30)
\end{aligned}
$$

Finally, we apply the QR factorization to express $\sqrt{r}B = QR$ where R is of size $a \times d$ where $a \leq d$. Then $rB'B = R'R = K$ and the required approximation to the right factor of the variance integral W in (4) is $H = RN_{1,1}'^{-1}$.

Note that in the construction of the blocks B_k in (26) the coefficients $w_{k,t}$ need be computed once and for all, given the choice of order q of the Padé approximation. Also, when applying the filter to an irregularly sampled series represented by a stationary model, the quantities $G(T')^t$ in (26) need only be computed once for application throughout the length of the series, with the time interval r being the only varying quantity.

The squaring method for computing the Padé approximation may be extended to the computation of the right factor H. Compute the approximation $M_K \approx \exp(r_K T)$, where $r_K = r/(2^K)$ and the corresponding approximation of the right factor H_K for r_K. Then the sequence of squaring and QR factoring

$$
M_k = M_{k+1}^2, \qquad H_k = Q \begin{pmatrix} H_{k+1} \\ H_{k+1}\, M_{k+1}' \end{pmatrix} \quad \text{for} \quad k = K-1 \ldots 0 \qquad (31)
$$

gives $M = M_0$ and $H = H_0$ as required, because $H_k' H_k = H_{k+1}' H_{k+1} + M_{k+1} H_{k+1}' H_{k+1} M_{k+1}'$ furnishes the integral (4) over the range 0 to r_k as the sum of the integrals from 0 to r_{k+1} and r_{k+1} to r_k.

4 The eigen-value method and comparisons

Let the transition matrix T have the eigenvalue decomposition

$$T = LDL^{-1} \tag{32}$$

where D is diagonal with elements d_i, the eigenvalues of T. This decomposition is always possible if the eigenvalues of T are distinct. If some are equal it may not be possible, and this may occur as a consequence of the design of some systems represented by the state equation. This is a clear limitation on the scope of the eigen-value method, but when the elements d_i are distinct it can be efficient and accurate, with resort to the Padé method when they are not distinct. When (32) is possible, the variance integral (4) can be expressed as

$$W = L \int_0^r \exp(hD) L^{-1} V (L')^{-1} \exp(hD') dh \, L'. \tag{33}$$

Setting $Q = L^{-1}V(L')^{-1}$, the integral in (33) can be evaluated explicitly as P where

$$P_{i,j} = Q_{i,j} E_{i,j}, \quad \text{where} \quad E_{i,j} = \frac{\exp\{r(d_i + \bar{d}_j)\} - 1}{(d_i + \bar{d}_j)}. \tag{34}$$

giving $W = LPL'$. Again, given the state space model, the decomposition (32) need only be derived once for application throughout the length of a time series, with (34) evaluated for varying values of r.

Choleski factorization can then be used to obtain the right factor H of W, but this may also be constructed by using the right factor G of V, avoiding some possible loss of numerical accuracy which the use of V itself may entail. This is done by factorization of the matrix E with elements $E_{i,j}$ in (34), as $E = U'U$. Because $E_{i,j}$ is close to r for small values of $r(d_i + \bar{d}_j)$, this can lead to near singularity of E which should be detected by the factorization procedure, and appropriate zero values set in U. Then define the matrix A consisting of vertically stacked block matrices A_k for $k = 1 \dots d$ (the state dimension) with elements given by

$$(A_k)_{i,j} = (GL'^{-1})_{i,j} U_{k,j} \tag{35}$$

We confirm that

$$\begin{aligned}
(A'A)_{\ell,j} &= \sum_{k=1}^{d} \sum_{i=1}^{m} (GL'^{-1})_{i,\ell} \, (GL'^{-1})_{i,j} \, U_{k,\ell} \, U_{k,j} && (36) \\
&= (L^{-1}G'GL'^{-1})_{\ell,j} \, (U'U)_{\ell,j} && (37) \\
&= Q_{\ell,j} E_{\ell,j} && (38) \\
&= P_{\ell,j}. && (39)
\end{aligned}$$

The final step is to apply QR factorization to reduce A to Z such that $Z'Z = P$, and the required right factor of W is $H = PL'$.

We have compared MATLAB implementations of the various methods described, for their accuracy and timings. The state space representation was derived for a CZAR model of order 4 for a process of dimension 3, giving a state dimension of 12. The results are given for an intermediate value of the time step r, over which 3 of the modes had decayed only by about 0.9, 5 by about 0.2, and the remaining 4 by values close to zero. Much smaller and larger values of r were also tried, to test out the reliability of the methods.

The shortest time to compute and factorize the variance integral (4) was the eigen-value method presented in (33) and (34). Taking this as the time unit for comparison, the method based directly on using the exponential in (5), formed using the built-in matrix exponential function, was timed at 2.5 units. The direct Padé approximation method for the variance factor, computed as described in section 3, was timed at 5 units and the direct eigen-value method for the variance factor was the same.

For the variance factors all methods agreed numerically to within 10^{-13}, except for the method based directly on using the exponential in (5), which differed from all these by 10^{-8}. For the variance, computed directly or from the factor, all methods agreed numerically to within 10^{-14}, except for the method based directly on using the exponential in (5), which differed from all these by 10^{-9}.

This latter method appears to be relatively inaccurate, and not the speediest, though twice as fast as the Padé method for the factor. The eigen-value method for the variance, followed by factorization, is speediest by a good factor of 5, and accurate in this example. However, it can not always be applied when there are equal eigen-values. The factorization step can also lead to loss of accuracy in badly conditioned examples. In the example used all the correlations of the 3×3 model disturbance matrix V_e were greater than 0.99, suggesting poor conditioning, and the eigen-value method for the variance still produced an accurate answer after factorization. However, on setting the second row of the factor G_e of V_e to zero, which made a difference of less than 0.5% to the lower right 2×2 part of V_e, the discrepancy between the Padé method and the factor from the eigen-value method for the variance increased to 10^{-9}, although their corresponding variances agreed to 10^{-14}. The two methods to agree to within 10^{-13} were the direct Padé approximation for the variance factor and the direct eigen-value method for the variance factor. These have the same timings, but given the potential difficulties with the latter, when eigen-values coincide, we prefer the former of these as the most reliable and accurate overall, though with some penalty factor on the speed of computation.

Appendix: matrix exponentials

We derive the result (5) by the explicit solution of a differential equation, using D for the differential operator. It is readily verified by differentiating the series

$$\exp(tH) = \sum_{k=0}^{\infty} (tH)^k / k! \tag{40}$$

that the solution to the differential equation

$$DX(t) = HX(t) \tag{41}$$

with $X(t) = X_0$ at $t = 0$, is

$$X(t) = \exp(tH)X_0. \tag{42}$$

Now partition

$$X(t) = \begin{pmatrix} y(t) \\ x(t) \end{pmatrix} \quad \text{and} \quad H = \begin{pmatrix} A & B \\ 0 & C \end{pmatrix}. \tag{43}$$

We show that the formal solution

$$\begin{pmatrix} y(t) \\ z(t) \end{pmatrix} = \exp \begin{pmatrix} tA & tB \\ 0 & tC \end{pmatrix} \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} \tag{44}$$

becomes explicitly

$$\begin{pmatrix} y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} \exp(tA) & \exp(tA) \int_0^t \exp(-sA) B \exp(sC) ds \\ 0 & \exp(tC) \end{pmatrix} \begin{pmatrix} y_0 \\ z_0 \end{pmatrix}. \tag{45}$$

Write (41) as the coupled equations

$$\begin{aligned} Dy &= Ay + Bz \\ Dz &= Cz. \end{aligned} \tag{46}$$

From the second equation

$$D\{\exp(-tC)z\} = \exp(-tC)Dz - C\exp(-tC)z = 0, \tag{47}$$

and integrate to obtain

$$[\exp(-sC)z]_0^t = \exp(-tC)z - z_0 = 0 \quad \Rightarrow \quad z(t) = \exp(tC)z_0. \tag{48}$$

From the first equation

$$\begin{aligned} D\{\exp(-tA)y\} &= \exp(-tA)Dy - A\exp(-tA)y \\ &= \exp(-tA)Bz = \exp(-tA)B\exp(tC)z_0 \end{aligned} \tag{49}$$

$$[\exp(-sA)y]_0^t = \exp(-tA)y - y_0 = \int_0^t \exp(-sA)B\exp(sC)dtz_0 \tag{50}$$

$$y(t) = \exp(tA)y_0 + \exp(tA)\int_0^t \exp(-sA)B\exp(sC)dsz_0. \tag{51}$$

From (48) and (51) we have (45). For the result (5) set $A = -T$, $B = V$ and $C = T'$.

References

G. Tunnicliffe Wilson, M. Reale, and J. Haywood. *Models for dependent time series.* New York, CRC Press, 2015.