# Model Solution: Numerical Algorithms

## Table of Contents



## Introduction

Macroeconomic models become more and more complex nowadays and, as a rule, are not tractable analytically. These models, in general, are solved numerically. Below we describe several techniques that are employed in Python Framework to find numerical solution.

## A. Non-Linear System

We are solving a system of equations,

$f(y) = 0$, where (A.1)

$$f(x_{t-1}, x_t, x_{t+1}, e_t) = \begin{cases} g_1(x_{t-1}, x_t, Ex_{t+1}, e_{1,t}) \\ \dots \\ g_N(x_{t-1}, x_t, Ex_{t+1}, e_{N,t}) \end{cases} \quad \text{and} \quad y_t = \begin{cases} x_{t-1} \\ x_t \\ x_{t+1} \end{cases}$$

Here $E$ is the expectation operator. The boundary conditions are:

$$x_0 = starting\ values; x_{T+1} = steady\ state\ values$$

Equations (A.1) are general equations for variables with a maximum lead and lag of one. If equations have lead and lags variables more than one, than these equations can be rewritten in the form of (1) by introducing new variables. For example, if equations have a variable $x_{t+2}$, we can introduce a new variable y, add new equation $y_t = x_{t+1}$ and rewrite original equations in form of (1).

We apply an iterative algorithm and linearize equations (A.1). At iteration k:

$$f\left(x_{t-1}^k, x_t^k, Ex_{t+1}^k\right) + \frac{\partial f^k}{\partial x_{t-1}}\left(x_{t-1}^{k+1} - x_{t-1}^k\right) + \frac{\partial f^k}{\partial x_t}\left(x_t^{k+1} - x_t^k\right) + \frac{\partial f^k}{\partial x_{t+1}}\left(Ex_{t+1}^{k+1} - Ex_{t+1}^k\right) = 0 \qquad \text{(A.2)}$$

These equations are linear with respect to next iteration variables, $x_{t-1}^{k+1}, x_t^{k+1}, x_{t+1}^{k+1}$. Equations (A.2) can be rewritten as,

$$L_t \Delta x_{t-1} + C_t \Delta x_t + F_t E \Delta x_{t+1} = -f_t \qquad \text{(A.3)}$$

Jacobians $L_t, C_t, F_t$ can be stacked and equations (A.3) can be represented in a matrix form:

$$\begin{pmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L_1 & C_1 & F_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_2 & C_2 & F_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & L_3 & C_3 & F_3 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & L_{T-2} & C_{T-2} & F_{T-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & L_{T-1} & C_{T-1} & F_{T-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & L_T & C_T & F_T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} \Delta x_0 \\ \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ . \\ \Delta x_{T-2} \\ \Delta x_{T-1} \\ \Delta x_T \\ \Delta x_{T+1} \end{pmatrix} = - \begin{pmatrix} 0 \\ f_1 \\ f_2 \\ f_3 \\ . \\ f_{T-2} \\ f_{T-1} \\ f_T \\ 0 \end{pmatrix} \qquad \text{(A.4)}$$

The size of the matrix is $N(T+2)$ by $N(T+2)$. Inverting this matrix could be problematic for large number of equations N or large time horizon T. Please note that this matrix is sparse. We can use sparse matrices linear algebra scipy package to solve these equations.

Equations (A.4) are solved iteratively until solution converges.

Another approach that can be used is an application of LBJ method. It is briefly described below. The starting values are, $\Delta x_0 = 0$. By substituting this value in (A.3) we get,

$\Delta x_1 = M_1 \Delta x_2 + d_1$, where $M_1 = -C_1^{-1} F_1$ and $d_1 = -C_1^{-1} f_1$.

Repeating these steps, we can find expressions:

$$M_t = -(C_t + L_t M_{t-1})^{-1} F_t$$
$$d_t = -(C_t + L_t M_{t-1})^{-1} (f_t + L_t d_{t-1}) \quad \text{(A.5)}$$

After computing these matrices, the solution can be easily obtained by backward substitution:

$$\Delta x_t = M_t \Delta x_{t+1} + d_{t,} \quad \Delta x_{T+1} = steady\ state \quad \text{(A.6)}$$

While the initial conditions are fixed and $\Delta x_0 = 0$, the terminal conditions are set to steady state solution. This requires one to find a steady state solution of system (A.1). A reasonable assumption would be to assume that the terminal conditions are floating and that the solution does not change in time for the right boundary, i.e., $\Delta x_{T+1} = \Delta x_T$. Then by substituting it in (A.6), we can find formula for the terminal condition,

$$\Delta x_{T+1} = (I - M_T)^{-1} d_T \quad \text{(A.7)}$$

The LBJ algorithm consists of two iterative steps: firstly, compute matrices $M$ and vector $d$ and secondly, compute solution $x$. These steps are repeated until the numerical solution converges.

Yet another assumption is that the endogenous variables at right boundary have fixed values. Then the terminal condition at the right boundary is, $\Delta x_{T+1} = 0$. Below we describe a modified LBJ method. By substituting this expression in (A.3) results in expression

$$\Delta x_T = M_T \Delta x_{T-1} + d_T, \text{ where } M_T = -C_T^{-1} L_T \text{ and } d_T = -C_T^{-1} f_T \quad \text{(A.8)}$$

The solution can be easily obtained by forward substitution:

$$\Delta x_{t+1} = M_{t+1} \Delta x_t + d_{t+1} \ ; \ \Delta x_0 = 0 \quad \text{(A.9)}$$

# B. Linear System

Equations (A.1) are linear with respect to endogenous and exogenous variables. These equations can be rewritten as,

$$L x_{t-1} + C x_t + \mathrm{F} x_{t+1} = -\mathrm{f} - \psi e_t \quad \text{(B.1)}$$

By stacking endogenous variables $x_t, x_{t-1}$ we can rewrite (3) as,

$$Ay_{t+1} = By_t + C + \psi e_t\,, \qquad y_t = \begin{cases} x_t \\ x_{t-1} \end{cases}$$
$$A = \begin{pmatrix} \mathrm{F} & 0 \\ 0 & I \end{pmatrix}, B = \begin{pmatrix} -\mathrm{C} & -\mathrm{L} \\ I & 0 \end{pmatrix}, C = \begin{pmatrix} -\mathrm{f} \\ 0 \end{pmatrix}, \Psi = \begin{pmatrix} -\psi \\ 0 \end{pmatrix} \tag{B.2}$$

Here A, B, C, $\psi$ are the constant matrices, $y_t$ are the endogenous variables and $e_t$ are the shocks.

## Christopher Sim's method

Christopher Sims showed that (B.2) can be translated into a system,

$$y_{t+1} = \theta_1 y_t + \theta_C + \theta_0 e_t + \theta_y \sum_{k=1}^{\infty} \theta_f^k \theta_z E(e_{t+k}) \tag{B.3}$$

For un-expected future shocks, the endogenous variable $y_{t+1}$is determined only by the current shock: $e_t$ , and equation (B.3) simplifies:

$$y_{t+1} = \theta_1 y_t + \theta_C + \theta_0 e_t \tag{B.4}$$

Contrary, all future anticipated shocks, govern the dynamics of endogenous variables and equation (B.3) holds.

## Binder and Pesaran Method

According to Binder & Pesaran (1997), if the unique and stable solution exists, it is given by:

$$x_t = \mathrm{B}\, x_{t-1} + H\mathrm{f} + \sum_{i=0}^{\infty} H^i \psi\, E(e_{t+i}) \tag{B.5}$$

Where $H = C(C + FB)^{-1}$ and $B$ satisfies a quadratic equation:

$$\mathrm{F}\,\mathrm{B}^2 + \mathrm{C}\,\mathrm{B} + \mathrm{L} = 0 \tag{B.6}$$

Authors employ iteration technique to solve quadratic equation. In general, this equation can have many solutions, a unique solution, or no solution at all.

## Anderson-Moore Method

Anderson-More algorithm requires no special treatment for models with multiple lags and leads. This distinguishes AIM method from all others where one should introduce new variables and cast the model in a form with at most one lead and one lag. For one lead and one lag model equations authors provide a simplified solution in the form:

$$x_t - x_0 = \mathrm{B}\,(x_{t-1} - x_0) + \sum_{i=0}^{\infty} F^i \Phi\, \psi\, E(e_{t+i}) \tag{B.7}$$

Here B is the reduced form coefficients matrix, Φ is the exogenous shock scaling matrix, and F is the exogenous shock transfer matrix. Anderson claim that AIM procedure exhibits significant computational performance for large scale models in terms of CPU time.

## Jaromír Beneš Method

This method uses QZ matrix factorization. Vector of endogenous variables $x_t$ can be partitioned into predetermined part $x_t^P$ and non-predetermined part $x_{t+1}^N$:

$$x_t = \begin{cases} x_t^P \\ x_{t+1}^N \end{cases} \tag{B.8}$$

Then equations (B.1) can be recast in the form of (B.2). Following Klein (1997), the system matrices A,B are decomposed by applying a Generalized Schur algorithm:

$$A = QTZ^H$$
$$B = QSZ^H \tag{B.9}$$

Here Q and Z are the unitary matrices, $Z^H$ is the conjugate transpose of matrix Z, and T, S are upper triangular matrices. Following Michal Andrle (2007), when shocks are not anticipated, the system of equations (B.2) has a solution in the state-space form:

$$\begin{pmatrix} x_t^N \\ \alpha_t \end{pmatrix} = \begin{pmatrix} T^F \\ T^A \end{pmatrix} \alpha_{t-1} + \begin{pmatrix} R^F \\ R^A \end{pmatrix} e_t + \begin{pmatrix} K^F \\ K^A \end{pmatrix}$$
$$x_t^P = \mathrm{U}\ \alpha_t \tag{B.10}$$

The transient matrix T, the shock matrix R and the constant vector K are given below:

$$\begin{aligned}
T^F &= Z_{21} \\
T^A &= -T_{11}^{-1}\, S_{11} \\
R^F &= (Z_{21}G + Z_{22})\, R^U \\
R^A &= -T_{11}^{-1}\, [\psi_1 + (S_{11}G + S_{12})\, R^U\,] \\
G &= -Z_{11}^{-1}\, Z_{12} \\
R^U &= -S_{11}^{-1}\, \psi_2 \\
U &= Z_{11} \\
K_u &= (T_{22} + S_{22})^{-1}\, C_2 \\
X_{a0} &= T_{11}^{-1}\, (S_{11}G + S_{12}) \\
X_{a1} &= G + T_{11}^{-1}\, T_{12} \\
K^F &= -(Z_{21}G + Z_{22})\, K_u \\
K^A &= -(X_{a0} + X_{a1})K_u - T_{11}^{-1}\, C_1
\end{aligned} \tag{B.11}$$

Here indices 1 and 2 denote part of matrices T, R and vector K that correspond to predetermined and non-predetermined transition variables.

When shocks are anticipated, solution (B.10) is augmented with the future shocks:

$$\begin{pmatrix} x_t^N \\ \alpha_t \end{pmatrix} = T\,\alpha_{t-1} + R\begin{pmatrix} e_t \\ e_{t+1} \\ \dots \\ e_{t+N} \end{pmatrix} + \begin{pmatrix} K^F \\ K^A \end{pmatrix} \quad \text{(B.12)}$$

Here R is matrix of the current and the future shocks:

$$R = \begin{bmatrix} R^F & X^F R^U & X^F J\,R^U & X^F J^2\,R^U & \dots & X^F J^{N-1}\,R^U \\ R^A & X^F R^U & X^A J\,R^U & X^A J^2\,R^U & \dots & X^A J^{N-1}\,R^U \end{bmatrix} \quad \text{(B.13)}$$

The auxiliary vectors and the matrix are shown below:

$$\begin{aligned} X^F &= Z_{21}G + Z_{22} \\ X^A &= X_{a1} + J\ X_{a0} \\ J &= -S_{22}^{-1}\,T_{22} \end{aligned} \quad \text{(B.14)}$$

# C. Judgmental Adjustments

In many cases user may have her or his view on a path of endogenous variables. Here we briefly describe the methodology that can be used to forecast variables with both anticipation and without anticipation.

Suppose that shocks at times $t, t+1, \dots, t+N$ are anticipated. Then, we can write at time $t+1$:

$$x_{t+1} = T\,x_t + K + R_0 e_t + R_1 e_{t+1} + R_2 e_{t+2} + \cdots + R_N e_{t+N} \quad \text{(C.1)}$$

At time $t+2$:

$$x_{t+2} = T\,x_{t+1} + K + R_0 e_{t+1} + R_1 e_{t+2} + R_2 e_{t+3} + \cdots + R_{N-1} e_{t+N}\ , \quad \text{(C.2)}$$

Or,

$$x_{t+2} = T\,(T\,x_t + K + R_0 e_t + R_1 e_{t+1} + R_2 e_{t+2} + \cdots + R_N e_{t+N}) + K + R_0 e_{t+1} + R_1 e_{t+2} + R_2 e_{t+3} + \cdots + R_{N-1} e_{t+N}$$

And,

$$x_{t+2} = T^2 x_t + (T+I)K + TR_0 e_t + (TR_1 + R_0)e_{t+1} + (TR_2 + R_1)e_{t+2} + \cdots + (TR_2 + R_{N-1})e_{t+N}$$

By induction, we can derive that at time $t+k$ the following equation holds:

$$x_{t+k} = T^k x_t + (T^{k-1} + T^{k-2} + \cdots + I)K + S_0^k e_t + S_1^k e_{t+1} + S_2^k e_{t+2} + \cdots + S_N^k e_{t+N}$$

Or,

$$x_{t+k} \quad = T^k x_t + (I - T)^{-1}(I - T^k)K + \sum_{i=0}^{N} S_i^k e_{t+i} \tag{C.3}$$

Here the aggregated shock matrix $S$ is:

$$S_i^k = \begin{cases} T\, S_i^{k-1}, & if\ k > i \\ T\, S_i^{k-1} + R_{i-k}, & if\ k \le i \end{cases} \qquad \text{and} \qquad S_i^0 = R_i \tag{C.4}$$

One can then solve equations (C.3) for values of the future shocks to match the path of endogenous variables to the desired level. In other words, the future shocks are "endogenized" and the corresponding variables are "exogenized". This numerical procedure can be optimized in terms of CPU memory and speed.

Suppose that user has a specific view on the path of endogenous variables which is given by, $\hat{x}_{t+k}$. Then, writing equation (C.3) for the deviation of the endogenous variables from this path, one can find adjustments to the future shocks $\Delta e_{t+k}$:

$$x_{t+k} - \hat{x}_{t+k} = S_1^k \Delta e_{t+1} + S_2^k \Delta e_{t+2} + \cdots + S_N^k \Delta e_{t+N} \tag{C.5}$$

By solving equation (C.5), one can find values of the new shock $\hat{e}_{t+k} = e_{t+k} + \Delta e_{t+k}$. This shock brings path of $x_{t+k}$ to the desired level of $\hat{x}_{t+k}$ .

Equation (C.5) simplifies when there is one un-expected shock at time $t + k$:

$$x_{t+k} - \hat{x}_{t+k} = R_0\, \Delta e_{t+k} \tag{C.6}$$

It can be easily solved. The new value of the shock is:

$$\hat{e}_{t+k} = e_{t+k} + R_0^{-1}(x_{t+k} - \hat{x}_{t+k}) \tag{C.7}$$

If there are several un-expected shocks, the new shock is computed at the first occurrence of this shock and the endogenous variables are forecasted onwards, then the shock is computed at the second occurrence of shock and the forecast is updated. This procedure is successively repeated until all shocks are accounted.

# D. Blanchard-Kahn Condition

Below we formulate condition for an existence of a unique solution of system of equations (A.1). We linearize these equations around a steady state solution $\acute{x}$:

$$A \begin{pmatrix} \Delta x_{t+1} \\ \Delta x_t \end{pmatrix} = B \begin{pmatrix} \Delta x_t \\ \Delta x_{t-1} \end{pmatrix} \tag{C.1}$$

In general, matrix A is a singular matrix and may not have its inverse. Because of that we cannot proceed with finding eigen values of $A^{-1}B$.

To find eigen values of equation (C.1) we apply QZ decomposition of matrices A, B. The eigen values can be expressed via diagonal elements of matrices T and S:

$$\lambda_i = \begin{cases} S_{i,i} * T_{i,i}^{-1}, if T_{i,i} \neq 0 \\ +\infty, if T_{i,i} = 0, S_{i,i} > 0 \\ -\infty, if T_{i,i} = 0, S_{i,i} < 0 \end{cases} \quad \text{(C.2)}$$

Eigen values (C.2) could be greater or less than one. Because of that some path of endogenous variables starting from the steady-state solution may diverge and some may converge. This is a typical saddle point instability problem. Blanchard and Kahn formulated a condition for existence and uniqueness of a model solution. It states that solution of equation (A.1) exists and is unique if and only if the number of eigen values $|\lambda_i| > 1$ is equal to the number of non-predetermined endogenous variables. In other words, the number of unstable eigen roots should be equal to the number of forward-looking endogenous variables (variables with leads greater than zero).

# E. Kalman Filter and Smoother

## Standard Filter and Smoother

Kalman Filter is an algorithm that uses a series of measurements observed over time containing statistical noise. It produces estimates of unknown variables that are more accurate than the estimates that are based on a law of motion of these variables alone. It assumes that underlying model is a state-space model,

$$x_{t+1} = F_{t+1}x_t + C_{t+1} + R_{t+1}w_{t+1}$$
$$z_{t+1} = H_{t+1}x_{t+1} + v_{t+1} \quad \text{(E.1)}$$

Here: t is the time index, $x_t$ are the state variables vector (unobserved values), $z_t$ are the measurement variables vector (observed values), $C_t$ is the vector of constants, $w_t$ and $v_t$ are the process and the measurement noises. The latter are assumed to be drawn from normal distribution:

$$w_t \sim N(0, Q_t)$$
$$v_t \sim N(0, R_t) \quad \text{(E.2)}$$

The covariance matrices Q and R are assumed to be diagonal and time independent:

$$Q_t = \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_N^2 \end{pmatrix} \quad \text{(E.3)}$$

$$R_t = \begin{pmatrix} \varepsilon_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \varepsilon_N^2 \end{pmatrix}$$

We introduce variables $\hat{x}_{t+1,t}$ and $P_{t+1,t}$ which are *a posteriori* state estimate of endogenous variables and of covariance matrix at time $t+1$ given observations up to and including at time $t$ .

Kalman filter is a two-step algorithm. In the prediction step, the state variables and posterior error covariance matrix P are estimated based on information available at the previous time,

$$\begin{aligned} \hat{x}_{t+1,t} &= F_{t+1}\hat{x}_{t,t} + C_{t+1} \\ P_{t+1,t} &= F_{t+1,t}P_{t,t}F_{t+1,t}^T + R_{t+1}Q_{t+1}R_{t+1}^T \end{aligned} \quad \text{(E.4)}$$

In the update step, the state variables are corrected based on measurements,

$$\begin{aligned} \hat{x}_{t+1,t+1} &= \hat{x}_{t+1,t} + K_{t+1}\left(z_{t+1} - H_{t+1}\hat{x}_{t+1,t}\right) \\ S_{t+1} &= H_{t+1}P_{t+1,t}H_{t+1}^T + R_{t+1} \\ K_{t+1} &= P_{t+1,t}H_{t+1}^T S_{t+1}^{-1} \\ P_{t+1,t+1} &= (I - K_{t+1}H_{t+1})P_{t+1,t} \end{aligned} \quad \text{(E.5)}$$

Here $K_{t+1}$ is the Kalman gain matrix, and $S_{t+1}$is the pre-fit residual covariance matrix. The prediction (E.4) and correction (E.5) steps are repeated to get the filtered unobservable variables for the entire time domain.

Kalman filter allows to compute filtered variables based on measurements available up to current time t. By using Kalman filter recursively one can compute filtered variables up to final time T. These filtered variables can be additionally rectified based on the whole set of measurements up to time T. These filtered variables are updated by applying Rauch-Tung-Striebel Smoother in a backward pass:

$$\begin{aligned} L_t &= P_{t,t}F_t^T P_{t+1,t}^{-1} \\ \hat{x}_{t,T} &= \hat{x}_{t,t} + L_t\left(\hat{x}_{t+1,T} - \hat{x}_{t+1,t}\right) \\ P_{t,T} &= P_{t,t} + L_t\left(P_{t+1,T} - P_{t+1,t}\right)L_t^T \end{aligned} \quad \text{(E.6)}$$

# Durbin and Koopman Approach

Durbin and Koopman proposed a multivariate and univariate approach to state filtering and smoothing. Below we briefly describe this method. According to this approach, the state equation is that shock is assumed to happen at the previous time step.

$$\begin{aligned} x_{t+1} &= F_t x_t + C_t + R_t w_t \\ z_t &= H_t x_t + v_t \end{aligned} \quad \text{(E.7)}$$

## Non-Diffuse Filter and Smoother

### Multivariate approach

The Kalman filter recursions for non-diffuse state multivariate filtering are:

$$\begin{aligned}
&\hat{x}_{t+1,} = F_t\hat{x}_t + C_t + K_t\varepsilon_t \\
&P_{t+1} = F_tP_tL_t^T + R_tQ_tR_t^T \\
&K_t = F_tP_tH_t^TS_t^{-1} \\
&\varepsilon_t = z_t - H_t\hat{x}_t \\
&L_t = F_t - K_tH_t \\
&S_t = H_tP_tH_t^T + R_tQ_tR_t^T
\end{aligned} \tag{E.8}$$

And the Kalman smoother recursions are:

$$\begin{aligned}
&r_{t-1} = H_t^TS_t^{-1}\varepsilon_t + L_t^Tr_t \\
&\tilde{x}_t = \hat{x}_t + P_tr_{t-1}
\end{aligned} \tag{E.9}$$

These recursions are applied backward for $t = N, N-1, \dots, 1$ with $r_N = 0$.

### Univariate approach

When measurements are precise and measurements errors are zeros, the matrix $S_t$ could become singular and its inverse can not be found. Durbin and Koopman proposed a univariate approach to state variables filtering. When the covariance matrices of the measurements errors and state variables errors are diagonal, the filtering equations for each of the components of filtered endogenous variable $\hat{x}_t = \left[\hat{x}_{t,1}\,;\ \hat{x}_{t,2}\,;\dots;\ \hat{x}_{t,n}\right]$ become:

$$\begin{aligned}
&\hat{x}_{t,i+1} = \hat{x}_{t,i} + M_{t,i}S_{t,i}^{-1}\varepsilon_{t,i} \\
&P_{t,i+1} = P_{t,i} - M_{t,i}S_{t,i}^{-1}M_{t,i}^T \\
&M_{t,i} = P_{t,i}H_{t,i}^T \\
&\varepsilon_{t,i} = z_{t,i} - H_{t,i}\hat{x}_{t,i} \\
&S_{t,i} = H_{t,i}P_{t,i}H_{t,i}^T + R_{t,i}Q_{t,i}R_{t,i}^T
\end{aligned} \tag{E.10}$$

The transition from time $t$ to time $t+1$ is achieved by relations

$$\begin{aligned}
&\hat{x}_{t+1,1} = F_t\hat{x}_{t,n} + C_t \\
&P_{t+1,1} = H_tP_{t,n}H_t^T + R_tQ_tR_t^T
\end{aligned}$$

In case $S_{t,i} = 0$, we have:

$$\begin{aligned}
&\hat{x}_{t,i+1} = \hat{x}_{t,i} \\
&P_{t,i+1} = P_{t,i}
\end{aligned}$$

The univariate smoother recursion is like the multivariate one:

$$
\begin{aligned}
r_{t,i-1} &= H_{t,i}^T S_{t,i}^{-1} \varepsilon_{t,i} + L_{t,i}^T r_{t,i} \\
r_{t-1,n} &= F_{t-1}^T r_{t,1} \\
\tilde{x}_t &= \hat{x}_t + P_t r_{t-1}
\end{aligned}
\tag{E.9}
$$

## Diffuse Filter and Smoother

Kalman filter and smoother require one to set initial condition for a posterior estimate of endogenous variable and its covariance matrix. Durbin and Koopman proposed an elegant way to treat these conditions when one does not have a full information and hence assumes that that initial covariance is infinite. The authors proposed a diffuse filtering and smoothing algorithm for state variables.

### Multivariate approach

The mean square error covariance matrix is decomposed into a finite $P_{*,t}$ an infinite $P_{\infty,t}$ parts: $P_t = P_{*,t} + kP_{\infty,t}$ . Authors derive asymptotic equations to the first order of $k^{-1}$ when $k \to \infty$.

$$
\begin{aligned}
\hat{x}_{t+1} &= F_t \hat{x}_t + C_t + K_{*,t} \varepsilon_t \\
P_{\infty,t+1} &= F_t P_{\infty,t} F_t^T \\
P_{*,t+1} &= F_t P_{*,t} L_t^T + R_t Q_t R_t^T \\
K_{*,t} &= F_t P_t H_t^T S_{*,t}^{-1} \\
\varepsilon_t &= z_t - H_t \hat{x}_t \\
L_{*,t} &= F_t - K_{*,t} H_t \\
S_t &= H_t P_{*,t} H_t^T + R_t Q_t R_t^T
\end{aligned}
\tag{E.10}
$$

These equations are applied for the first $d$ periods until the infinite part of covariance matrix $P_\infty$ becomes zero, i.e., $P_{\infty,d} = 0$. Thereafter the non-diffusive state filtering algorithm (E.8) is used. State smoothing backward recursions start at the last period $N$. Regular algorithm (E.9) is applied until period $d$, and then it is followed by a diffuse state smoothing one:

$$
\begin{aligned}
\tilde{x}_t &= \hat{x}_t + P_{*,t} r_{t-1}^{(0)} + P_{\infty,t} r_{t-1}^{(1)} \\
r_{t-1}^{(0)} &= L_\infty^T r_t^{(0)} \\
r_{t-1}^{(1)} &= H_t^T (S_{\infty,t}^{-1} \varepsilon_t - K_{*,t}^T r_t^{(0)}) + L_{\infty,t}^T r_t^{(1)}
\end{aligned}
\tag{E.11}
$$

with initialization $r_d^{(0)} = r_d$ and $r_d^{(1)} = 0$

### Univariate approach

The diffuse state filtering recursions are:

$$\begin{aligned}
&\hat{x}_{t,i+1} = \hat{x}_{t,i} + K_{\infty,t,i} S_{\infty,t,i}^{-1} \varepsilon_{t,i} \\
&P_{*,t,i+1} = P_{*,t,i} + K_{\infty,t,i} K_{\infty,t,i}^{T} S_{*,t,i} S_{\infty,t,i}^{-2} - \left( K_{*,t,i} K_{\infty,t,i}^{T} + K_{\infty,t,i} K_{*,t,i}^{T} \right) S_{\infty,t,i}^{-1} \\
&P_{\infty,t,i+1} = P_{\infty,t,i} - K_{\infty,t,i} K_{\infty,t,i}^{T} S_{\infty,t,i}^{-1}
\end{aligned} \tag{E.12}$$

In the case where $S_{\infty,t,i} = 0$, the usual filtering equations apply,

$$\begin{aligned}
&\hat{x}_{t,i+1} = \hat{x}_{t,i} + K_{*,t,i} S_{*,t,i}^{-1} \varepsilon_{t,i} \\
&P_{*,t,i+1} = P_{*,t,i} + K_{*,t,i} K_{*,t,i}^{T} S_{*,t,i}^{-1} \\
&P_{\infty,t,i+1} = P_{\infty,t,i}
\end{aligned} \tag{E.13}$$

The transition from time $t$ to time $t+1$ is achieved by,

$$\begin{aligned}
&\hat{x}_{t+1,1} = F_t \hat{x}_{t,n} + C_t \\
&P_{*,t+1,1} = F_t P_{*,t,n} F_t^{T} + R_t Q_t R_t^{T} \\
&P_{\infty,t+1,1} = F_t P_{\infty,t,n} F_t^{T}
\end{aligned} \tag{E.14}$$

The diffuse smoothing backward recursions start at period $d$,

$$\begin{aligned}
&\tilde{x}_t = \hat{x}_t + P_{*,t} r_{t-1}^{(0)} + P_{\infty,t} r_{t-1}^{(1)} \\
&r_{t,i-1}^{(0)} = L_{\infty}^{T} r_{t,i}^{(0)} \\
&r_{t,i-1}^{(1)} = H_{t,i}^{T} S_{\infty,t}^{-1} \varepsilon_t + L_{*,t,i}^{T} r_{t,i}^{(0)} + L_{\infty,t,i}^{T} r_{t,i}^{(1)} \\
&L_{\infty,t,i} = I - K_{\infty,t,i} H_{t,i} S_{\infty,t,i}^{-1} \\
&L_{*,t,i} = \left( K_{\infty,t,i} S_{*,t,i} S_{\infty,t,i}^{-1} - K_{*,t,i} \right) H_{t,i} S_{\infty,t,i}^{-1}
\end{aligned} \tag{E.15}$$

with initialization $r_{d,n}^{(0)} = r_d$ and $r_{d,n}^{(1)} = 0$

# F. Initial Conditions

Kalman filter requires initial conditions for a posterior estimate of state variables and error variances. There are several options to set these conditions. For example, one can take historical data to set these initial values,

$$\begin{aligned}
&\tilde{x}_0 = x_0 \\
&\tilde{P}_0 = P_0
\end{aligned} \tag{F.1}$$

Another option would be to use asymptotic values that can be derived from state-space model equations. The state variables (E.1) can be decomposed into stable and unstable parts. The stable states correspond to the absolute values of eigen values of transition matrix $F_t$ less than one, and the unstable states – to greater than one. One can assume that initial condition for the estimate of unstable states are zeros, while for the stable states are defined by asymptotic conditions,

$$\tilde{x}_{0,unstable} = 0$$
$$\tilde{x}_{0,stable} = (I - F_\infty)^{-1} C_{stable} \quad \text{(F.2)}$$

Similarly, we can write equations for an estimate of error covariance matrices:

$$P_{\infty,0} = 0$$
$$P_{*,0} = F_\infty P_{*,0} F_\infty^T + R_\infty Q_\infty R_\infty^T \quad \text{(F.3)}$$

The latter equation is a discrete Lyapunov type equation. It can be rewritten in the form,

$$(I_{n^2} - F_\infty \otimes F_\infty)\, vec\left(P_{*,0}\right) = vec(R_\infty Q_\infty R_\infty^T) \quad \text{(F.4)}$$

Here $\otimes$ is the Kronecker product operator, and $vec$ is the vector operator. The latter stacks columns of a matrix on top of each other. Equation (F.4) has a solution of the form,

$$vec\left(P_{*,0}\right) = (I_{n^2} - F_\infty \otimes F_\infty)^{-1}\ vec(R_\infty Q_\infty R_\infty^T) \quad \text{(F.5)}$$

# G. Model Calibration

It is a common practice to assume that probability of model fit can de decomposed into prior probability of model parameters $\psi_1(p)$ and conditional probability of series $x_t$ given observations $y_t$ and parameters $p$, $\psi_2\left(x_t|\, x_{t-1,}x_{t-2}, \dots, x_0, p\right)$. The latter is assumed to be a Markov process, i.e., $\psi_2 = \psi_2\left(x_t|\, x_{t-1,}, p\right)$. A reasonable assumption is that these probabilities are independent, so that log of a product of probabilities is additive:

$$log(\psi) = log(\psi_1) + log(\psi_2) \quad \text{(G.1)}$$

Here $\psi$ is the probability of model fit to the data. The probability $\psi_1$is a prior probability and depend on user assumptions about its distribution. For example, user may assume that the prior probability has Normal , Lognormal, Beta, Gamma, Inverse Gamma, Student t, Weibull, Wishart, etc., density functions.

Assuming normal pdf, $\psi_1$ has the form:

$$log(\psi_1) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{n} \frac{1}{\sigma_i} (p_i - \bar{p}_i)^2 \quad \text{(G.2)}$$

Here $\bar{p}_i$ is the mean of parameter, and $\sigma_i$ is it standard deviation.

Following Bayesian interpretation of Kalman filter, the distribution of probability $\psi_2$ can be written as,

$$log(\psi_2) = -\frac{N_y \log(2\pi)}{2} - \frac{1}{2} \sum_{t=1}^{T} \log(\det(S_t)) - \frac{1}{2} \sum_{t=1}^{T} \sum_{i}^{n} \left(x_{t,i} - y_{t,i}\right)^2 \quad \text{(G.3)}$$

Here $N_y$ is the dimension of measurement vector $y$.

Calibration of model parameters consists in searching for parameters $p$ that maximize likelihood $\psi$ given equations (G.1) and (G.2)

$$max(\psi(p|\bar{p}, \sigma, y_t, y_{t-1}, , , y_1)) \quad \text{(G.4)}$$

# E. Parameters Sampling

We apply a **Markov chain Monte Carlo** (MCMC) algorithm to create random samples of model parameters. These random draws of parameters are characterized by a probability density function that tends to the distribution (G.1) as the number of draws increases. There are multiple algorithms such as acceptance-rejection Metropolis-Hastings method which draws continuous random variable given its probability density function, Gibs sampling which requires all the condition distributions of (G.1) to be sampled, Hamilton Monte Carlo method which applies Hamilton dynamics, and many more.

These algorithms require a lot of steps to converge for skewed probability distributions. For example. a model could be highly sensitive to changes in one parameter and not sensitive to changes in another parameter. The framework uses algorithm of Jonathan Goodman and Jonathan Weare (2010). This algorithm falls under umbrella of MCMC algorithm. It is affine invariant ensemble sampler algorithm which works well for skewed distributions.

# References


Anderson-Moore Algorithm, https://www.federalreserve.gov/econres/ama-index.htm

Durbin J., Koopman S.J. (2002). "A Simple and Efficient Simulation Smoother for State Space Time Series Analysis." Biometrika, 89(3), 603–615. doi:10.1093/biomet/89.3.603.

Durbin J., Koopman S.J. (2012). Time Series Analysis by State Space Methods. 2nd edition. Oxford University Press, New York.

S.J. Koopman and J. Durbin (1998). "Fast Filtering and Smoothing for Multivariate State Space Models." Journal of Time Series Analysis, Vol. 21, No. 3.

S.J. Koopman and J. Durbin (1999). "Filtering and Smoothing of State Vector for Diffuse State-Space Models." Journal of Time Series Analysis, Vol. 24, No. 1.

Michal Andrle (2007). "Linear Approximation to Policy Function in IRIS Toolbox". https://michalandrle.weebly.com/uploads/1/3/9/2/13921270/michal_iris_solve.pdf.

Christopher Sims (1996). "Solving Rational Expectations Models." Seminar Paper, http://sims.princeton.edu/yftp/gensys/LINRE3A.pdf

Klein P. (1997): “Using the Generalized Schur Form to Solve a System of Linear Expectational Difference Equations,” discussion paper, IIES, Stockholm University, klein@iies.su.se.

Michael Binder and M. Hashem Pesaran. (1997). “Multivariate rational expectations models. Characterization of the Nature of the Solutions and Their Fully Recursive Computation.” https://www.jstor.org/stable/pdf/3532897.pdf?refreqid=excelsior%3A5e82c86594ec2406dc1607c3c337e0

Sebastien Villemot (2011). “Solving Rational Expectations Models at First Order: What Dynare Does.” https://www.dynare.org/wp-repo/dynarewp002.pdf

Jonathan Goodman and Jonathan Weare (2010), "Ensemble Samplers with Affine Invariance". https://projecteuclid.org/download/pdf_1/euclid.camcos/1513731992

“Kalman Filter”, https://en.wikipedia.org/wiki/Kalman_filter

“Lyapunov Equation”, https://en.wikipedia.org/wiki/Lyapunov_equation

“Schur decomposition”, https://en.wikipedia.org/wiki/Schur_decomposition

“Markov chain Monte Carlo”, https://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo

“Metropolis–Hastings algorithm”, https://en.wikipedia.org/wiki/Metropolis%E2%80%93Hastings_algorithm

“Gibbs sampling”, https://en.wikipedia.org/wiki/Gibbs_sampling

“Hamilton Monte Carlo”, https://en.wikipedia.org/wiki/Hamiltonian_Monte_Carlo