



# PM Note: state space representation in Dynare

DATUM: 2017-02-24  
AVDELNING:  
HANDLÄGGARE: Ingvar Strid  
HANTERINGSKLASS Ö P P E N

SVERIGES RIKSBANK  
SE-103 37 Stockholm  
(Brunkebergstorg 11)

Tel +46 8 787 00 00  
Fax +46 8 21 05 31  
registratorn@riksbank.se  
www.riksbank.se

---

DNR [Diarienummer]

## State space representation of a model implemented in Dynare, in the original ordering of the variables in the model

This note describes how to construct the state space representation of a model implemented with Dynare 4.3.0. (Note that it does not work for all versions of Dynare due to changes in the `M_` and `oo_` structures between versions.) When setting up a model in Dynare the user defines the endogenous variables in the model after the `var` command. The innovations in the model are defined after the `varexo` command. The order of the variables given in the code corresponds to the order in `M_endo_names` and `M_exo_names` respectively. We refer to this as the *original ordering*.

The state-space representation of a model consists of i) a state equation and ii) an observation equation. This note only concerns the state equation,

$$X(t)=F*X(t-1)+B0*e(t)$$

which is described by the state transition matrix, `F`, and the matrix multiplying the innovations in the state equation, `B0`. The state vector `X` contains all the variables in the model (defined under `var`) and the vector `e` contains all the innovations (defined under `var_exo`). We are interested in constructing the matrices `F` and `B0`. (The notation is used by Warne for YADA and I think it goes back to Hamilton's 'Time Series Analysis'...). The observation equation is easily constructed. In Dynare one includes the observation equations among the state equations, and when applying commands the observed variables are listed.

We have:

- $m=\dim(X)=M\_endo\_nbr$
- $n=\dim(e)=M\_exo\_nbr$

The reason we need this note is that in solving the model Dynare does two things which makes it harder to follow what is going on concerning representations. It a) reshuffles the

variables in  $X$  and  $b$ ) compresses the vector  $X$  to contain only the 'true states'.<sup>1</sup> Hence, the matrices  $F$  and  $B$  are not immediately available.

Dynare delivers the solution of the model (just an alternative way of representing the state equation) on the format

$$X_a(t) = F_a * X_b(t) + B_a * e(t)$$

where  $X_a$  is a re-ordering of  $X$ , and  $\dim(X_a) = m$ , and where  $X_b$  is both a re-ordering and compression of  $X$ , only containing the true state variables.

We have:

- $F_a = \text{oo\_dr.ghx}$
- $B_a = \text{oo\_dr.ghu}$

Now, based on  $(F_a, B_a)$  we want to retrieve the state-space representation with the original ordering of the variables,  $(F, B_0)$ .

The re-shuffling of variables in the vector  $X$  are described by the index vectors:

- $\text{oo\_dr.order\_var}$ : re-ordering from  $X$  to  $X_a$
- $\text{oo\_dr.inv\_order\_var}$ : re-ordering from  $X_a$  to  $X$

The indices of the true state variables are given in  $M\_state\_var$ , which is with respect to the variables as they are ordered in  $X_a$ . The true state variables appear in compressed form in  $X_b$ . We re-construct this index vector such that it gives the indices of the true state variables in  $X$ , instead of in  $X_a$ . This index vector is named **stateOrderNew**. (This is the 'hard part').

Once this has been done we can construct  $F$  and  $B$  as:

- $F(:, stateOrderNew) = \text{oo\_dr.ghx}(\text{oo\_dr.inv\_order\_var}, :);$
- $B_0 = \text{oo\_dr.ghu}(\text{oo\_dr.inv\_order\_var}, :);$

We now have the state-space model in the original ordering,  $(F, B_0)$ .

Finally, this produces an 'interface' with the Klein solver in YADA which delivers the solution on exactly the form  $(F, B_0)$  described above. If we work with the model in both Dynare and YADA it should facilitate easier comparison of solutions. Also, it allows us to make changes to the model and the code in Dynare and simply import them in the YADA Klein solver function, 'Dynare within YADA'.

---

<sup>1</sup> A true state is characterised by a corresponding non-zero column in  $F$ . However, I'm not really sure that Dynare actually computes the minimal state. I am not sure it is actually the 'true state' vector, but the main point is that the dimension of the compressed state vector,  $X_b$ , is significantly reduced compared with the user-defined vector of variables,  $X$ .



## Code: computeStateSpaceRepresentation.m

Written in 2013. Might be good to have a look at how Dynare has changed since then.

```
-----  
  
numberOfStates=M_.endo_nbr;  
numberOfInnovations=M_.exo_nbr;  
  
F_=zeros(numberOfStates,numberOfStates);  
B0_=zeros(numberOfStates,numberOfInnovations);  
  
% F_(:,M_.state_var)=oo_.dr.ghx(oo_.dr.inv_order_var,:);  
  
% numberOfDynareStateVars: 'Effective' or 'real' number of state  
variables (economic states)  
  
numberOfDynareStateVars=size(M_.state_var,2);  
stateOrderNew=zeros(numberOfDynareStateVars,1);  
iter=0;  
for i=1:M_.endo_nbr  
    stateVarInOrderVar=oo_.dr.order_var(i,1);  
    for j=1:size(M_.state_var,2)  
        stateVarInStateVar=M_.state_var(1,j);  
        if stateVarInOrderVar==stateVarInStateVar  
            iter=iter+1;  
            stateOrderNew(iter,1)=stateVarInOrderVar;  
        end;  
    end;  
end;  
F_(:,stateOrderNew)=oo_.dr.ghx(oo_.dr.inv_order_var,:);  
B0_=oo_.dr.ghu(oo_.dr.inv_order_var,:);
```