Determinacy and Classification of Markov-Switching Rational Expectations Models: A Technical Guide

Seonghoon Cho*

June 17, 2019

Abstract

This is a technical guide for the MOD (minimum of modulus) method proposed by Cho (2019) in the class of general Linear rational expectations (LRE) models and Markov-switching rational expectations (MSRE) models. This document provides a compact summary of the paper and illustrates how to implement the methodology. Current versions of the codes accompanying this technical guide and the numerical examples including the ones in the paper can be found at https://sites.google.com/site/sc719en. This technical guide and the matlab codes will be updated in the future.

*School of Economics, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul, 120-749, Korea. *E-mail:* `sc719@yonsei.ac.kr`. Tel:+82-2-2123-2470; Fax:+82-2-393-1158

1 Introduction

This document explains how to implement the MOD(minimum of modulus) method of Cho (2019) using matlab for LRE models and matlab combined with a mathematics language called "Singular" developed by Decker et al. (2019) for MSRE models. The proposed method uses only the most mean-square stable MOD solution for a complete classification of models into three mutually disjoint and exhaustive subsets: determinacy, indeterminacy and the case of no stable solution. A solution methodology for the MOD solution is also provided. Previous technical guide for Cho (2016) is simplified and nested in this note.

Table of Contents

Appendix

2 MOD Method for LRE Models

2.1 LRE Models and RE solutions

$$x_t = AE_t[x_{t+1}] + Bx_{t-1} + Cz_t, \tag{1a}$$

$$z_t = Rz_{t-1} + \epsilon_t, \qquad \epsilon_t \sim i.i.d(0, D) \tag{1b}$$

x_t $n \times 1$ vector of endogenous variables,
z_t $m \times 1$ vector of mean-square stable exogenous variables,
ϵ_t $m \times 1$ vector of white noises,
A $n \times n$ coefficient matrix of the forward-looking endogenous variables,
B $n \times n$ coefficient matrix of the backward-looking endogenous variables,
C $n \times m$ coefficient matrix of the exogenous variables,
R $m \times m$ coefficient matrix of the lagged exogenous variables.

Writing a model into the form (1a): Any linear rational expectations model can be written as $B_1 x_t = A_1 E_t[x_{t+1}] + B_2 x_{t-1} + C_1 z_t$ where B_1, A_1, B_2 and C_1 by collecting the parameters of current, forward-looking, backward-looking endogenous variables and exogenous variables. Therefore, one can set $A = B_1^{-1} A_1$, $B = B_1^{-1} B_2$ and $C = B_1^{-1} C_1$.

Solution Forms, Restrictions and Full Set of MSV solutions

$$x_t = [\Omega x_{t-1} + \Gamma z_t] + w_t : \text{General RE solution}, \tag{2}$$

$$x_t = \Omega x_{t-1} + \Gamma z_t : (w_t = 0_{n\times 1}) : \text{MSV solution}, \tag{3}$$

$$w_t = FE_t w_{t+1} : (w_t \neq 0_{n\times 1}) : \text{Sunspot component}. \tag{4}$$

$$\Omega = (I_n - A\Omega)^{-1} B, \tag{5}$$

$$\Gamma = (I_n - A\Omega)^{-1} C + F\Gamma R, \tag{6}$$

$$F = (I_n - A\Omega)^{-1} A. \tag{7}$$

$$\mathcal{S} = \left\{ \Omega \in \mathcal{C}^{n\times n} \middle| \; r(\Omega^1) \leq r(\Omega^2) \leq ... \leq r(\Omega^N) \right\} \tag{8}$$

where $r(M) = \max_{1\leq i \leq n}(|\xi_i|)$, $\xi_1, ..., \xi_n$ are the eigenvalues of an $n \times n$ matrix M.

Definition 1 *$x_t = \Omega^1 x_{t-1}$ is an MOD solution if $r(\Omega^1) = \min r(\Omega)$ for all $\Omega \in \mathcal{S}$.*

2.2 The MOD Method

2.2.1 Classification of LRE Models by the *MOD* Method

The main classification result of the *MOD* method for LRE models is summarized in Result 3 of the paper.

Result 3 *Consider a LRE model (1) and the set of solutions $\mathcal{S}$ in (8). Then, necessary and sufficient conditions for determinacy, indeterminacy and the case of no stable solution are given in Table (1). Moreover, if there exists a real-valued solution such that $r(\Omega)r(F) < 1$, it is the unique MOD solution and the model is determinacy-admissible.*

Table 1: **Classification of LRE Models by the MOD Method**

	Determinacy-Admissible(DA): Ω^1 is real and $r(\Omega^1)r(F^1) < 1$	Determinacy-Inadmissible(DIA): Ω^1 is complex or $r(\Omega^1)r(F^1) \geq 1$
Determinacy	$r(\Omega^1) < 1$, $r(F^1) \leq 1$	Impossible
Indeterminacy	$r(F^1) > 1$	$r(\Omega^1) < 1$
No Stable Solution	$r(\Omega^1) \geq 1$	$r(\Omega^1) \geq 1$

2.2.2 Identification of the *MOD* Solution

The *MOD* methodology is completed by finding the *MOD* solution. In LRE models, identification of the *MOD* can be done by use of the eigensystem implied by the generalized Schur decomposition theorem. Such an identification is not possible for MSRE models because of the lack of an eigensystem. In principle, identification of the *MOD* can be done by solving all Ω, which is very difficult even for small-scale LRE models, let alone MSRE models. Partitioning the LRE models by Determinacy-Admissibility(DA) provide a crucial identification condition for the *MOD* solution. Implementation will be shown below.

2.3 Standard Methods Using Eigensystem

This section shows the equivalence between the MOD approach and the standard methods based on an eigensystem. This also helps understand our results in terms of well-known generalized eigenvalue. Let $y_t = [x_t' \; x_{t-1}']'$. Model (1a) can be reformulated as

$$\tilde{B}y_t = \tilde{A}E_t y_{t+1} + \tilde{C}z_t \tag{9}$$

$$\tilde{B} = \begin{bmatrix} I_n & -B \\ I_n & 0_{n\times n} \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A & 0_{n\times n} \\ 0_{n\times n} & I_n \end{bmatrix}, \tilde{C} = \begin{bmatrix} C \\ 0_{n\times m} \end{bmatrix} \tag{10}$$

Let $\xi_{A,B}$ be the set of generalized eigenvalues implied by the model. Formally,

$$\xi_{A,B} = \{\xi_i \in \mathcal{C} | 0 = |\tilde{A} - \xi_i \tilde{B}|, \; |\xi_1| \leq |\xi_i| \leq |\xi_{2n}| \; \} \tag{11}$$

Any solution $\Omega \in \mathcal{S}$ is associated with n out of $2n$ eigenvalues. The corresponding F is associated with the inverses of the remaining n eigenvalues.

Using these properties, LRE models can also be classified as Table 2, which is exactly the same as Table 1, including the conditions for Determinacy-Admissibility(DA) and Determinacy-Inadmissibility(DIA). Detailed arguments for equivalence of the classification of LRE models using this standard eigensystem approach and the MOD method can be found in the paper.

Table 2: **Classification of LRE Models**

	Determinacy-Admissible(DA): $\Omega(\xi_1, ..., \xi_n) \in S$ and $|\xi_n| < |\xi_{n+1}|$	Determinacy-Inadmissible(DIA): $\Omega(\xi_1, ..., \xi_n) \notin S$ or $|\xi_n| = |\xi_{n+1}|$
Determinacy	$|\xi_n| < 1 \leq |\xi_{n+1}|$	Impossible
Indeterminacy	$|\xi_{n+1}| < 1$	$r(\Omega^1) = |\xi_{n+i}| < 1, \, i \geq 0$
No Stable Solution	$|\xi_n| \geq 1$	$r(\Omega^1) = |\xi_{n+i}| \geq 1, \, i \geq 0$

Remark 1. Table 2 shows that a usual root-counting to identify determinacy can fail but only when $\Omega(\xi_1, ..., \xi_n) \notin \mathcal{S}$. Therefore, the existence of the MOD solution must be checked along with root-counting, which is precisely what the MOD method does in line with the gensys algorithm of Sims (2002).

2.4 Implementing MOD Method for LRE Models

MOD method is to classify a given model into DET/INDET/NSS by identifying and computing the MOD solution. There are two main matlab codes for LRE models to be explain in the following section.

- fmlre.m: Implement the MOD method using the modified forward method.
- qzmlre.m Implement the MOD method using eigensystem.

There are two ways of implementing the method. First, QZ method does it all. This computes the MOD solution using the eigensystem. Therefore, one can directly classify a given model using $r(\Omega^1)r(F^1)$ following Table 1. Second, a sequential way of first using the forward method and QZ method if necessary. The forward method computes the forward solution Ω^*. If this turns out to be the MOD solution, then classification is done. If it does not exist or $r(\Omega^*)r(F^*) \geq 1$ and $r(\Omega^*) < 1$, the forward method cannot tell classification. Thus one needs to use an alternative technique to identify the MOD solution and complete classification. Here QZ method is suggested because the eigensystem directly yields the MOD solution. Note that in the MSRE models, QZ method is not available. Thus one must use a sequential approach of applying the forward method first and use the Gröbner basis approach if necessary. The sequential approach is not needed for LRE models, but it shows how it would work for MSRE models.

1. Implementing the MOD method using the QZ method.

   (a) Write your model in the form of (1a), e.g., "Ex.m". Construct A. Construct B, C, or R if any.

   (b) Type and run DETCMOD=qzmlre(A,B). Ω^1 is the MOD solution where DETCMOD=$[r(\Omega^1)\ r(F^1)\ r(\Omega^1)r(F^1)]$.

      i. If $r(\Omega^1)r(F^1) < 1$, then the model is DA. $r(\Omega^1)$ and $r(F^1)$ tells whether the model is DET/INDET/NSS. Done.

      ii. If $r(\Omega^1)r(F^1) \geq 1$ and $r(\Omega^1) < 1$, the model is indeterminate. (The model is DIA)

      iii. If $r(\Omega^1)r(F^1) \geq 1$ and $r(\Omega^1) \geq 1$, the model has no stable solution. (The model is DIA)

2. Implementing the MOD method using the forward method

   (a) Same as 1.a.

   (b) Type and run DETC=fmlre(A,B). Ω^* is the forward solution where DETC=$[r(\Omega^*)\ r(F^*)\ r(\Omega^*)r(F^*)]$.

      i. If $r(\Omega^*)r(F^*) < 1$, then $\Omega^* = \Omega^1$, and the model is DA. $r(\Omega^1)$ and $r(F^1)$ tells whether the model is DET/INDET/NSS. Done.

      ii. If $r(\Omega^*)r(F^*) \geq 1$ and $r(\Omega^*) < 1$, the model is indeterminate. Done.

      iii. If $\Omega^* = NaN$, forward solution does not exists. Apply the alternative to find the MOD solution and complete classification.

      iv. If $r(\Omega^*)r(F^*) \geq 1$ and $r(\Omega^*) \geq 1$, the forward method cannot tell the classification. Apply the alternative to find the MOD solution and complete classification.

Remark 1. Only the case i would arise in practice because cases ii, iii and iv would arise for DIA models. Therefore, forward method would also be sufficient for classification of economic models.

Remark 2. In fact, forward method is even faster than standard methods in many cases.

3 MOD Method for MSRE Models

3.1 MSRE Models and RE solutions

The class of MSRE models:

$$x_t = E_t[A(s_t, s_{t+1})x_{t+1}] + B(s_t)x_{t-1} + C(s_t)z_t, \quad (12)$$

$$z_t = R(s_t)z_{t-1} + \epsilon_t, \qquad \epsilon_t \sim (0, D) \quad (13)$$

x_t	$n \times 1$ vector of endogenous variables,
z_t	$m \times 1$ vector of mean-square stable exogenous variables,
ϵ_t	$m \times 1$ vector of white noises,
s_t	S-regime ergodic Markov chain,
P	$S \times S$ transition matrix with $p_{ij} \equiv \Pr(s_t = j \mid s_{t-1} = i)$, $i, j \in \{1, 2, ..., S\}$,
$A(\cdot), B(\cdot)$	$n \times n$ coefficient matrices,
$C(\cdot)$	$n \times m$ coefficient matrix,
$R(\cdot)$	$m \times m$ coefficient matrix.

Writing a model into the form (12): The original model would be written as $B_1(s_t)x_t = E_t[A_1(s_t, s_{t+1})x_{t+1}] + B_2(s_t)x_{t-1} + C_1(s_t)z_t$. Therefore, one can set $A(s_t, s_{t+1}) = B_1^{-1}(s_t)A_1(s_t, s_{t+1})$, $B(s_t) = B_1^{-1}(s_t)B_2(s_t)$ and $C(s_t) = B_1^{-1}(s_t)C_1(s_t)$. Both Cho (2016) and Foerster2016QE emphasize that A may well depend on future regime s_{t+1} in DSGE models with microfoundation subject to regime-switching. Using the perturbation method, Foerster et al. (2016) derive a general form of linearized Markov-switching DSGE models, which can also be written as (12).

Solution Forms and Restrictions

$$x_t = [\Omega(s_t)x_{t-1} + \Gamma(s_t)z_t] + w_t : \text{General RE solution}, \quad (14)$$

$$x_t = \Omega(s_t)x_{t-1} + \Gamma(s_t)z_t : (w_t = 0_{n\times 1}) : \text{MSV solution}, \quad (15)$$

$$w_t = E_t[F(s_t, s_{t+1})w_{t+1}] : (w_t \neq 0_{n\times 1}) : \text{Sunspot component} \quad (16)$$

$$\Omega(s_t) = \{I_n - E_t[A(s_t, s_{t+1})\Omega(s_{t+1})]\}^{-1}B(s_t), \quad (17)$$

$$\Gamma(s_t) = \{I_n - E_t[A(s_t, s_{t+1})\Omega(s_{t+1})]\}^{-1}C(s_t) + E_t[F(s_t, s_{t+1})\Gamma(s_{t+1})R(s_{t+1})], \quad (18)$$

$$F(s_t, s_{t+1}) = \{I_n - E_t[A(s_t, s_{t+1})\Omega(s_{t+1})]\}^{-1}A(s_t, s_{t+1}), \quad (19)$$

Complete Set of MSV solutions:

$$\mathcal{S} = \left\{ \Omega(s_t) \in \mathcal{C}^{n\times n} | r(\bar{\Psi}_{\Omega^1\otimes\Omega^1}) \leq r(\bar{\Psi}_{\Omega^2\otimes\Omega^2}) \leq ... \leq r(\bar{\Psi}_{\Omega^N\otimes\Omega^N}) \right\}, \tag{20}$$

where $\bar{\Psi}_{\Omega\otimes\Omega}$, and $\Psi_{F\otimes F}$ are defined as:

$$\begin{aligned}
\bar{\Psi}_{\Omega\otimes\Omega} &= \begin{bmatrix} p_{11}\Omega(1)\otimes\Omega(1) & ... & p_{S1}\Omega(1)\otimes\Omega(1) \\ ... & ... & ... \\ p_{1S}\Omega(S)\otimes\Omega(S) & ... & p_{SS}\Omega(S)\otimes\Omega(S) \end{bmatrix}, \\
\Psi_{F\otimes F} &= \begin{bmatrix} p_{11}F(1,1)\otimes F(1,1) & ... & p_{1S}F(1,S)\otimes F(1,S) \\ ... & ... & ... \\ p_{S1}F(S,1)\otimes F(S,1) & ... & p_{SS}F(S,S)\otimes F(S,S) \end{bmatrix}
\end{aligned}$$

The following matrices will also be needed for exposition.

$$\begin{aligned}
\Psi_{G\otimes H} &= [p_{ij}G(i,j)\otimes H(i,j)] \\
\bar{\Psi}_{G\otimes H} &= [p_{ji}G(j,i)\otimes H(j,i)]
\end{aligned}$$

where $G = G(s_t, s_{t+1})$ and $H = H(s_t, s_{t+1})$ are $n \times n$ matrices and the arguments of $\Psi_{G\otimes H}$ and $\bar{\Psi}_{G\otimes H}$ are ij-th $n^2 \times n^2$ block. Only the following two will be used for our classification result. But general forms above are needed for proofs.

Definition 2 *A MSV solution (15) is mean-square stable (MSS) if and only if* $r(\bar{\Psi}_{\Omega\otimes\Omega}) < 1$. *A Sunspot* w_t *in (16) has the form:* $w_t = \Lambda(s_{t-1}, s_t) + V(s_t)V'(s_t)\eta_t$ *and is MSS if and only if* $r(\bar{\Psi}_{\Lambda\otimes\Lambda}) < 1$.

Definition 3 $x_t = \Omega^1(s_t)x_{t-1}$ *is an MOD solution in the mean-square stability sense if* $r(\bar{\Psi}_{\Omega^1\otimes\Omega^1}) = \min r(\bar{\Psi}_{\Omega\otimes\Omega})$ *for all* $\Omega(s_t) \in \mathcal{S}$ *in (20).*

3.2 The MOD Method

3.2.1 Classification of MSRE Models by the *MOD* Method

The main classification result of the *MOD* method for MSRE models is summarized in Proposition 3 of the paper.

Proposition 3 *Consider an MSRE model (12) and the set of solutions $\mathcal{S}$ in (19). Then, necessary and sufficient conditions for determinacy, indeterminacy and the case of no stable solution in the mean-square stability sense are given in Table (3). Moreover, if a real-valued solution such that $r(\Omega)r(F) < 1$, it is the unique MOD solution and the model is determinacy-admissible.*

Table 3: **Classification of MSRE Models**

	Determinacy-Admissible(DA) $\Omega^1(s_t)$ is real-valued and $r(\bar{\Psi}_{\Omega^1\otimes\Omega^1})r(\Psi_{F^1\otimes F^1}) < 1$	Determinacy-Inadmissible(DIA) $\Omega^1(s_t)$ is complex-valued or $r(\bar{\Psi}_{\Omega^1\otimes\Omega^1})r(\Psi_{F^1\otimes F^1}) \geq 1$
Determinacy	$r(\bar{\Psi}_{\Omega^1\otimes\Omega^1}) < 1$, $r(\Psi_{F^1\otimes F^1}) \leq 1$	Impossible
Indeterminacy	$r(\Psi_{F^1\otimes F^1}) > 1$	$r(\bar{\Psi}_{\Omega^1\otimes\Omega^1}) < 1$
No Stable Solution	$r(\bar{\Psi}_{\Omega^1\otimes\Omega^1}) \geq 1$	$r(\bar{\Psi}_{\Omega^1\otimes\Omega^1}) \geq 1$

The following corollary is an important and new finding in the literature. LRE models with lagged variables, the uniqueness of a MSV solution implies determinacy. In stark contrast, it is not the case for MSRE models. Therefore a correct classification requires one to check not just uniqueness of a MSV solution but also non-existence of stable sunspots following Proposition 3.

Corollary 1 *The uniqueness of a mean-square stable MSV solution does not always imply determinacy in MSRE models with lagged variables. That is, a model can be determinacy even when $r(\bar{\Psi}_{\Omega^1\otimes\Omega^1}) < 1 \leq r(\bar{\Psi}_{\Omega^2\otimes\Omega^2})$ because this can be compatible with $r(\Psi_{F^1\otimes F^1}) > 1$.*

3.2.2 Implementing MOD Method for MSRE Models

The methodology is completed by finding the *MOD* solution. Partitioning the entire class of MSRE models by determinacy-admissibility(DA) and determinacy-inadmissibility(DIA) provide an important identification condition for the MOD solution. This is crucial in practical application. There are two main matlab codes for MSRE models to be explain in the following section.

- fmmsre.m : Implements the MOD method using the modified forward method.
- gbmsre.m (for Windows users) or gbmsre_mac.m (for Windows users) : Implements the MOD method using the Gröbner basis (GB).

GB technique should not be used as a first technique to identify the MOD. gbmsre.m (or gbmsre_mac.m) is provided to complete the MOD method theoretically. It would not be needed in practice because all economic models would be DA and the forward method would be self-sufficient. To use GB approach, one should install a program called Singular. See below. To implement the method, proceed with the following steps.

1. Write your model in the form of (12), e.g., "Ex.m". Construct P and A. Construct B, C or R if any.
2. Type and run DETC=fmmsre(P,A,B,..). Ω^1 is the MOD solution where DETC=$[r(\bar{\Psi}_{\Omega^*\otimes\Omega^*})\ r(\Psi_{F^*\otimes F^*})\ r(\bar{\Psi}_{\Omega^*\otimes\Omega^*})r(\Psi_{F^*\otimes F^*})]$.
   (a) If $r(\bar{\Psi}_{\Omega^*\otimes\Omega^*})r(\Psi_{F^*\otimes F^*}) < 1$, then $\Omega^*(s_t) = \Omega^1(s_t)$, and the model is DA. $r(\bar{\Psi}_{\Omega^*\otimes\Omega^*})$ and $r(\Psi_{F^*\otimes F^*})$ tells whether the model is DET/INDET/NSS. Done.
   (b) If $r(\bar{\Psi}_{\Omega^*\otimes\Omega^*})r(\Psi_{F^*\otimes F^*}) \geq 1$ and $r(\Omega^*) < 1$, the model is indeterminate. Done. One does not need to check whether Ω^* is the MOD solution.
   (c) If $\Omega^* = NaN$, forward solution does not exists.
   (d) If $r(\Omega^*)r(F^*) \geq 1$ and $r(\Omega^*) \geq 1$, the forward method cannot tell the classification.
3. If c or d occurs, use the Gröbner Basis approach. Refer to the implementation procedure below.

As we keep emphasizing, the forward method would be self-sufficient if one considers economic models, which would almost surely be determinacy-admissible. That is, one would encounter only Case a because the remaining cases would imply that the model at hand would be determinacy-inadmissible. Therefore, the GB approach would not be required in practice.

Implementing the Gröbner Basis. gbmsre.m or gbmsre_mac.m does not directly apply the Gröbner basis in matlab. Actual computation is done in the language called Singular.

For Windows Users

1. Preparation

   (a) Installation: Install Singular following a default option from https://www.singular.uni-kl.de/index.php/singular-download.html. (It will be installed at c:\cygwin64)

   (b) Opening Singular command window: Open Cygwin64 Terminal and type 'singular' in that command window.

2. Implementing the GB technique

   (a) Write your model in the form of (12), e.g., "Ex.m". Construct P , A and B.

   (b) Type and run DETCMOD=gbmsre(P,A,B)

   It will display the following message in matlab command window and the code will temporarily hold. Copy and Paste the following

   execute(read("GB_singular.ascii")); timer; ns;

   in Singular command window, and press the Enter key. Then, Singular will read GB_singular.ascii generated by gbmsre.m and compute all solutions. If a command prompt ">" shows up in Singular, computation is done. The first number shown is the computation time in milliseconds. The second is the number of solutions. Singular will produce output files Sol_partC.txt and Sol_partC.txt.

   (c) Come back to matlab window and press any key. Then "Ex.m" will load the Singular output files, transform them into the solution format in matlab, report the total number of solutions, the MOD and all other MSV solution, and the information in Table 3. The results can confirm the equivalence of the MOD solution and the forward solution, Proposition 3 and Corollary 1.

3. Interpretation of the Results and Completing Classification

   (a) Your code "Ex.m" will produce a table with

   DETC=$[r(\bar{\Psi}_{\Omega^i\otimes\Omega^i})\ r(\Psi_{F^i\otimes F^i})\ r(\bar{\Psi}_{\Omega^i\otimes\Omega^i})r(\Psi_{F^i\otimes F^i})\ r(\Psi_{\Omega^{i\prime}\otimes F^i})]$, $i = 1, ..., N$.

   (b) The first row is the result for the MOD solution. Follow Table 3 for classification.

For Mac Users

1. Preparation

   (a) Installation : Follow the instruction below carefully.

      i. Go to System Preferences and open Security & Privacy in your mac computer. Allow apps downloaded from "Open anyway". (These steps are required because otherwise Singular cannot be installed properly.)

      ii. Go to https://www.singular.uni-kl.de/index.php/singular-download.html. Choose OS X and Choose the option "Installation from a dmg File", and follow the instruction there.

   (b) Open Singular command window

2. Implementing the GB technique: Same as Windows except the following.

   (b) Type and run DETCMOD=gbmsre_mac(P,A,B)

   It will display the following message in matlab command window and temporarily holds. Copy and Paste the following

   execute(read("/Applications/GB_singular.ascii")); timer; ns;

   in Singular command window, and press the Enter key. ...

3. Same as Windows

Technical Note on Gröbner Basis.

For Windows Users: gbmsre,m produces a file GB_singular.ascii at C:\ cygwin64 \ home \ Computername where Computername is your computer name, which is automatically identified. The code transforms the restriction (17) as a set of polynomials and writes it in Singular language as GB_singular.ascii . Singular reads this ascii file, computes the Gröbner basis and stores all the solutions at the directory above as Sol_partC.txt and Sol_partC.txt. Your code "Ex.m" loads these output files, transforms them into matlab format and produces the results in matlab command window.

For Mac Users: gbmsre_mac.m does the same thing as above. The only difference is that the directory "Applications" is the location where GB_singular.ascii and output files are stored.

Appendix

A Set of Matlab Codes for LRE and MSRE Models

The package named as "MODmethodyyyymmdd" contains several folders.

1. MOD Method: Folder LRE and MSRE contain the following set of main codes and supplementary ones. The folder includes two main folders containing MOD method and examples for LRE models and MSRE models.

	LRE	MSRE
	Code	Code
Forwar Method	fmlre.m	fmmsre.m
Alternatives	qzmlre.m	gbmsre.m, gbmsre_mac.m
Supplementary	modlre_sample.m qzmlre2gensys.m methods_comparison_LRE.m	modmsre_sample.m gbmsre_Testing_computation_time.m gbmsre_Testing_computation_time_mac.m Example_for_Corollary1.m lambda_min.m lambda_min_Example.m

2. Examples. This folder contains three subfolders with some examples replicating the results of the three papers. Cho (2016), the present paper Cho (2019) and a companion paper Cho and Moreno (2019).

	Codes
2016RED	x1.m, x2.m, x2Q.m, x2figure.m, x3.m
2019MOD	ReplicatingFigure1.m, Example_Corollary1_Fisher_FTPL.m
2019FTPLZLB	To be updated

In what follows, main codes are explained in detail. Supplementary codes for the MOD methodology are briefly explained as it is self-explanatory. Finally, examples are explained.

A.1 MOD Method - LRE Models

A.1.1 Main Codes

qzmlre.m

This is the main code implementing the MOD method using the the eigensystem, the QZ decomposition. qzmlre stands for the QZ method for LRE models. The code adopts some routines such as reorder.m and qzswitch.m in the package of Sims (2002), but applies to our representation of LRE models.

[DETCMOD, OmegaMOD,GammaMOD,FMOD,Geig,gvAll,OmegaAll,GammaAll,FAll]=qzmlre(A,B,C,R)

- **Input Arguments**:

Input	format	Concepts		Default
A	$n \times n$ matrix	A	Required	
B	$n \times n$ matrix	B	Optional	B $= 0_{n\times n}$
C	$n \times m$ matrix	C	Optional	C $= I_{n\times n}$
R	$m \times m$ matrix	R	Optional	R $= 0_{n\times n}$

- **Output Arguments:**

Output	Format	Concepts
DETCMOD	1×3 vector	DETCMOD $= [r(\Omega^1) \;\; r(F^1) \;\; r(\Omega^1)r(F^1)]$
OmegaMOD	$n \times n$ matrix	OmegaMOD $=\Omega^1$ (MOD solution)
GammaMOD	$n \times n$ matrix	GammaMOD $=\Gamma^1$ (MOD solution)
FMOD	$n \times n$ matrix	FMOD $=F^1$ (MOD solution)
Geig	$2n \times 1$ vector	Vector of the generalized eigenvalues
gvAll	$N \times n$ matrix	$N = \#$ of MSV solutions. Each row is the order of Geig
OmegaAll	$N \times 1$ cell array	i-th solution is associated with i-th gvAll. $1 \le i \le N$
GammaAll	$N \times 1$ cell array	Same as above
FAll	$N \times 1$ cell array	Same as above

Note. If $r(\Omega^1)r(F^1) > 1$, then there is no solution associated with the n smallest generalized eigenvalues.

- **Interpreting the Results: Refer to Table** 1 or 2.

<u>fmlre.m</u>:

This is the main code implementing the MOD method using the modified forward method. fmlre stands for the forward method for LRE models.

[DETC, FCC, OmegaK,GammaK,FK,OtherOutput,IRS]=fmlre(A,B,C,R,Opt);

- **Input Arguments:** Same as those of qzmlre.m. In addition, there is an optional input, Opt. See the code for detail.
- **Output Arguments:**

Output	**Format**	**Concepts**
DETC	1×3 vector	DETC $= [r(\Omega^*) \;\; r(F^*) \;\; r(\Omega^*)r(F^*)]$
FCC	1×2 vector	FCC $= [K \;\; r(R' \otimes F^*)]$, K is # of forward iteration
OmegaK	$n \times n$ matrix	OmegaK $= \Omega^*$
GammaK	$n \times n$ matrix	GammaK $= \Gamma^*$ if FCC2 < 1 $= \Gamma$ if FCC2 > 1
FK	$n \times n$ matrix	FK $= F^*$
OtherOutput		Other output arguments: See the code for details
IRS	$T \times nm$ matrix	Impulse-response function with horizon T.

- Interpretation of DETC. Refer to Table 1

1. If DETC(3)=$r(\Omega^*)r(F^*) < 1$, then the model is determinacy-admissible and $\Omega^* = \Omega^{MOD}$.
2. If DETC(3)≥ 1 and $r(\Omega^*) < 1$, the model is indeterminate.
3. If DETC(3)≥ 1 and $r(\Omega^*) \geq 1$, or DETC(3)=NaN (Ω^* does not exist), then use qzmlre.m to find the MOD solution.

Remark 1. If FCC(2)< 1 , GammaK $= \Gamma^*$ and therefore, OmegaK and GammaK constitute the forward solution. If FCC(2) ≥ 1, the FCC condition is violated. Thus even if Ω^* exists, a correct interpretation is that there is no forward solution because Γ^* does not exist. The code still produces Γ using the formula (6), which is rejected as an RE equilibrium on the ground of the NBC criterion. Refer to Cho and McCallum (2015) for detail.[1]

A.1.2 Supplementary Codes

qzmlre2gensys.m

This code makes gensys algorithm of Sims (2002) comparable to qzmlre.m and fmlre.m. Specifically, for given matrices A,B and C in (1a), define

$$k_t = E_t x_{t+1},\ x_t = k_{t-1} + \eta_t \tag{21}$$

Then the model can be written as

$$\Gamma_0 \hat{y}_t = \Gamma_1 \hat{y}_{t-1} + \Pi z_t + \Psi \eta_t \tag{22}$$

where $\hat{y}_t = [x_t'\ k_t']$ and Γ_0, Γ_1, Π and Ψ are the input matrices of gensys algorithm as functions of A,B and C such that:

$$\Gamma_0 = \begin{bmatrix} I_n & -A \\ I_n & 0_{n\times n} \end{bmatrix},\ \ \Gamma_1 = \begin{bmatrix} B & 0_{n\times n} \\ 0_{n\times n} & I_n \end{bmatrix}, \Pi = \begin{bmatrix} C \\ 0_{n\times m} \end{bmatrix},\ \Psi = \begin{bmatrix} 0_{n\times n} \\ I_n \end{bmatrix}$$

It is conventional to define k_t manually to include only non-zero expectational variables in vector $\hat{y}_t$ so that the dimension of these matrices are less than $2n$ if A has a rank less than n. Although (22) has a larger dimension in general, it makes the transformation model-independent, thus one does not need to write a given model in the form of gensys input manually.

The gensys algorithm computes the generalized eigenvalues ξ with respect to the matrix pencil $[\Gamma_0 - \xi\Gamma_1]$ via the Schur decomposition theorem. It turns out that the generalized eigenvalues of $[\Gamma_0 - \xi\Gamma_1]$ are exactly the same as those of $[\tilde{A} - \xi\tilde{B}]$. The algorithm examines the existence of the unique stable solution by what is known as spanning conditions. The code has the following form:

[G1,C,impact,fmat,fwt,ywt,gev,eu,loose]=qzmlre2gensys(A,B,C,c,div)

- **Input Arguments**: A,B,C are the same as those of fmlre.m and qzmlre.m. c and div

[1] What it means by FCC(2)> 1 can be understood as follows. The situation is like FCC(2)=$|r| > 1$ where $\Gamma_k = \sum_{i=1}^{k} r^{k-1}$ which explodes thus does not exist. The formula (6) is like $1/(1-r)$, which is clearly wrong.

are optional input arguments of gensys.m.

- **Output Arguments:** Outputs are the same as those of gensys.m.

- **Interpreting the Results:** Follow the code gensys.m. A key result is the vector eu, which shows the existence and uniqueness. The conditions for the classification by the gensys and the corresponding conditions by the MOD solution can be stated as follows.

Classification	eu	MOD
Determinacy	[1 1]	$r(\Omega^1) < 1$, $r(F^1) \leq 1$
Indeterminacy	[1 0]	$r(\Omega^1) < 1$, $r(F^1) > 1$
No Stable Solution	eu(1) < 1	$r(\Omega^1) \geq 1$

The MOD conditions are those by pulling the DA and DIA partitions in Table (1).

modlre_sample.m

This code applies the three solution techniques to 7 atheoretical models belong to DA-DET, DA-INDET, DA-NSS, DIA-INDET(real- and complex-valued MOD solution) and DIA-NSS(real- and complex-valued MOD solution). This shows that qzmlre.m is a stand-alone procedure. The code also shows that fmlre.m is equivalent to qzmlre.m except for DIA model with complex-valued solutions. DIA models are not economic ones and taken from Sims (2007).

1. **Instruction** : Choose one of the seven models (See the instruction of the code.) and run it. This code automatically implement the forward method followed by the QZ approach using an auxiliary code modlre_FM_QZ.m. One can manually choose fmlre.m or qzmlre.m.

methods_comparison_LRE.m

This code shows the equivalence between QZ method (qzmlre.m), the sequential technique (fmlre.m and qzmlre.m if necessary) and gensys algorithm (qzmlre2gensys.m) for all 7 examples above. Simply run the code.

A.2 MSRE Models

A.2.1 Main Codes

fmmsre.m

This is the main matlab code implementing the MOD method using the modified forward method (explained below). It takes P, $A(s_t, s_{t+1})$ and optional $B(s_t), C(s_t)$ or $R(s_t)$ as input arguments and produces the following output: conditions for determinacy, indeterminacy or the case of no stable solution, forward convergence condition (FCC), the forward solution and others. The function format, input and main outputs are given in the following way.

[DETC, FCC, OmegaK,GammaK,FK,OtherOutput,IRS]=fmmsre(P,A,B,C,R,Opt);

where fmmsre stands for the forward method for MSRE models. (OmegaK=$\Omega^*(s_t)$, GammaK=$\Gamma^*(s_t)$, FK=$F^*(s_t, s_{t+1})$)

- **Input Arguments:**

Input	format	Concepts		Default
P	$S \times S$ matrix	P(i,j)=$P(s_{t+1} = j \| s_t = i)$	Required	
A	$S \times S$ cell array or $S \times 1$ cell array	A{i,j}=$A(s_t = i, s_{t+1} = j\}$ or A{i,1}=$A(s_t = i\}$ if $A(\cdot)$=$A(s_t)$	Required	
B	$S \times 1$ cell array	B{i,1}=$B(s_t = i\}$	Optional	B{i,1}= $0_{n\times n}$
C	$S \times 1$ cell array	C{i,1}=$C(s_t = i\}$	Optional	C{i,1}= $I_{n\times n}$
R	$S \times 1$ cell array or $m \times m$ matrix	R{i,1}=$R(s_t = i\}$ or R=R	Optional	R= $0_{n\times n}$
Opt	Other optional input arguments: See the code			

Remark 1. To conduct impulse-response analysis, we recommend to specify $C(s_t) = B_1^{-1}(s_t)C_1(s_t)$.

Remark 2. Specify optional inputs as [] if they are followed by other inputs. For instance, set B=[]; C=[]; to specify R and/or Opt.

- **Output Arguments:**

Output	Format	Concepts
DETC	1×4 vector	$[r(\bar{\Psi}_{\Omega^* \otimes \Omega^*})\ \ r(\Psi_{F^* \otimes F^*})\ \ r(\bar{\Psi}_{\Omega^* \otimes \Omega^*})r(\Psi_{F^* \otimes F^*})\ r(\Psi_{\Omega^{*\prime} \otimes F^*})]$
FCC	1×2 vector	$[K\ \ r(\Psi_{R' \otimes F^*})]$, K is # of forward iteration
OmegaK	$S \times 1$ cell array	OmegaK{i,1} $= \Omega^*(s_t)$
GammaK	$S \times 1$ cell array	GammaK{i,1} $= \Gamma^*(s_t)$ if FCC2 < 1 $= \Gamma(s_t)$ if FCC2 > 1
FK	$S \times S$ cell array	FK{i,j}$= F^*(s_t, s_{t+1})$
OtherOutput, IRS	Other output arguments: See the code for details	

- Interpretation of DETC. Refer to Table 3

1. If DETC(3)$=r(\bar{\Psi}_{\Omega^* \otimes \Omega^*})r(\Psi_{F^* \otimes F^*}) < 1$, then the model is determinacy-admissible and $\Omega^* = \Omega^{MOD}$.
2. If DETC(3)≥ 1 and $r(\bar{\Psi}_{\Omega^* \otimes \Omega^*}) < 1$, the model is indeterminate.
3. If DETC(3)≥ 1 and $r(\bar{\Psi}_{\Omega^* \otimes \Omega^*}) \geq 1$, or DET(3)=NaN ($\Omega^*(s_t)$ does not exist), then use gbmsre.m to find the MOD solution.

Remark 3. There is an additional condition in DETC, $r(\Psi_{\Omega^{*\prime} \otimes F^*})$ unlike the LRE models. Recall that DETC $= [r(\Omega^*)\ \ r(F^*)\ \ r(\Omega^*)r(F^*)]$ in LRE models where $r(\Omega^*)r(F^*) = r(\Omega^{*\prime} \otimes F^*)$. In MSRE models, $r(\bar{\Psi}_{\Omega^* \otimes \Omega^*})r(\Psi_{F^* \otimes F^*}) < 1$ implies that $r(\Psi_{\Omega^{*\prime} \otimes F^*}) < 1$ but the converse is not true. This is the major difference between the two types of models. This is precisely the reason why the uniqueness of a stable MSV solution does not mean determinacy in general. In fact, such a case can always be generated by the order-preserving transformation of the model as long as the MOD solution is stable in DIA model such that $r(\Psi_{\Omega^{1\prime} \otimes F^1}) < 1$ but $r(\bar{\Psi}_{\Omega^1 \otimes \Omega^1})r(\Psi_{F^1 \otimes F^1}) > 1$. Refer to Corollary 1.

Remark 4. If FCC(2)< 1, GammaK $= \Gamma^*(s_t)$ and therefore, OmegaK and GammaK constitute the forward solution. If FCC(2) ≥ 1, the FCC condition is violated. Thus there is no well-defined forward solution even if $\Omega^*(s_t)$ exists. The code still produces $\Gamma(s_t)$ using the formula (6), which is rejected as an RE equilibrium on the ground of the NBC criterion. Refer to Cho (2016) for detail.

gbmsre.m or gbmsre_mac.m

gbmsre.m is the main matlab code implementing the MOD method using the Gröbner basis approach in Windows. Mac users must use gbmsre_mac.m. It plays the same role as qzmlre.m for LRE models but identifies the MOD solution by computing all MSV solutions. It takes the same input P, $A(s_t, s_{t+1})$ and $B(s_t)$ as those of fmmsre.m where a non-zero $B(s_t)$ for at least one regime must be included. The code produces the following output: conditions for determinacy, indeterminacy or the case of no stable solution. The function format, input and main outputs are given in the following way.

[DETCMOD,OmegaMOD,FMOD,DETC_All,AllOmegas]=gbmsre(P,A,B);

where gbmsre stands for the Gröbner basis method for MSRE models. (OmegaMOD=$\Omega^1(s_t)$, FMOD=$F^1(s_t, s_{t+1})$).

Unlike fmmsre.m, this code must use the language called "Singular", which must be installed in advance. Therefore, one must manually follow the step displayed in the matlab command window.

- **Input Arguments:** same as those of fmmsre.m but P, A and B must be specified because the purpose of this code is to find all MSV solutions when lagged variables are present with $B \neq 0_{n\times n}$.
- **Output Arguments:**

Output	Format	Concepts
DETCMOD	1×5 vector	$[r(\bar{\Psi}_{\Omega^1\otimes\Omega^1})\ \ r(\Psi_{F^1\otimes F^1})\ \ r(\bar{\Psi}_{\Omega^1\otimes\Omega^1})r(\Psi_{F^1\otimes F^1})\ r(\Psi_{\Omega^{1\prime}\otimes F^1})\ \ rc]$ and $rc = 1$ if the solution Ω^1 is real-valued and 0 otherwise.
OmegaMOD	$S \times 1$ cell array	OmegaMOD{i,1} $= \Omega^1(s_t)$
FMOD	$S \times S$ cell array	FMOD{i,j}$= F^1(s_t, s_{t+1})$
DETC_All	$N \times 5$ cell array	DETC_All{i,:}= $[r(\bar{\Psi}_{\Omega^i\otimes\Omega^i})\ \ r(\Psi_{F^i\otimes F^1})\ \ r(\bar{\Psi}_{\Omega^i\otimes\Omega^i})r(\Psi_{F^i\otimes F^i})\ r(\Psi_{\Omega^{i\prime}\otimes F^i})\ rc(i)]$ for $i = 1, ...N$, where N is the total # of MSV solutions and $rc(i) = 1$ if Ω^i is real-valued and 0 otherwise.
AllOmegas	$N \times S$ cell array	AllOmegas{i,s}$=\Omega^i(s_t = s)$.

A.2.2 Supplementary Codes

All of the following examples contain detailed explanation in the code. Run each of them and follow the instruction.

lambda_min.m

For a given $F(s_t, s_{t+1})$, this code computes $\Lambda_m(s_t, s_{t+1})$ such that $r(\Psi_{F\otimes F})r(\bar{\Psi}_{\Lambda_m\otimes\Lambda_m}) = 1$ following the analytical form presented in Proposition 1 of Cho (2019). This is important because it states that there exists no mean-square stable sunspot components associated with a given $F(s_t, s_{t+1})$ if and only if $r(\Psi_{F\otimes F}) \leq 1$. This improves Lemma 2 of Cho (2016). This code just confirms that Proposition 1 holds numerically. In fact, the $r(\Psi_{F\otimes F})r(\bar{\Psi}_{\Lambda\otimes\Lambda}) \geq 1$ for all possible $\Lambda(s_t, s_{t+1})$ is the relation that Farmer et al. (2009) wants to verify this numerically but not completely. Proposition 1 proves their conjecture by providing an analytical form of $\Lambda_{\min}(s_t, s_{t+1})$.

The function format is given by

[Lmin,tau2,xi2,V,D,u,Q]=lambda_min(P,F)

- **Input Arguments:** The code takes P and F as input arguments.

Input	format	Concepts	Default
P	$S \times S$ matrix	P(i,j)=$P(s_{t+1} = j \mid s_t = i)$	Required
F	$S \times S$ cell array or $S \times 1$ cell array	F{i,j}=$F(s_t = i, s_{t+1} = j\}$ or F{i,1}=$F(s_t = i\}$ if $F(\cdot) = F(s_t)$	Required

- **Output Arguments:**

Output	Format	Concepts
Lmin	$S \times S$ cell array	Lmin{i,j} $= \Lambda_{\min}(s_t, s_{t+1}) = \arg\min r(\bar{\Psi}_{\Lambda\otimes\Lambda})$ s.t. $w_t = E_t[F(s_t, s_{t+1})w_{t+1}] = E_t[F(s_t, s_{t+1})\Lambda_{\min}(s_t, s_{t+1})w_t]$
tau2	*scalar*	tau2 $= \tau_2 = r(\bar{\Psi}_{\Lambda_{\min}\otimes\Lambda_{\min}})$, xi2 $= \xi_2 = r(\Psi_{F\otimes F})$
V	$S \times 1$ cell array	V $= V(s_{t+1})$ such that
Phi	$S \times S$ cell array	Phi $= \Phi(s_t, s_{t+1})$
		$\Lambda_{\min}(s_t, s_{t+1}) = V(s_{t+1})\Phi(s_t, s_{t+1})V'(s_t)$
D,u,Q		Refer to Proposition 1 of Cho (2019)

lambda_min_Example.m

Sample code for using lambda_min.m: Finding Λ_m is not required as it is proven in Proposition 1. This code just shows how the formula works numerically. Simply run it. If one has computed any solution $\Omega(s_t)$ and the corresponding $F(s_t, s_{t+1})$ to a model, just type and run lambda_min.m.

modmsre_sample.m

This code applies the three solution techniques to 7 atheoretical models belong to DA-DET, DA-INDET, DA-NSS, DIA-INDET(real- and complex-valued MOD solutions) and DIA-NSS(real- and complex-valued MOD solutions). The code also shows that fmmsre.m is equivalent to modmsre_FM_GB.m except for DIA model with complex-valued solutions.

- Choose one of the seven models (See the instruction of the code.) and run it. This code automatically implement the forward method followed by the GB approach using an auxiliary code modmsre_FM_GB.m. One can manually choose fmmsre.m or gbmsre.m.

Example_for_Proposition4.m

This code shows an example in which the model has a unique stable MSV solution but is indeterminate.. The sample model is randomly generated, which implies that this is not a rare event, but an intrinsic property of MSRE models. Just run it.

GBmsre_Testing_computation_time.m or **GBmsre_Testing_computation_time_mac.m**

This code shows that the computation time to compute all MSV solutions using the Gröbner basis approach increases exponentially as the dimension of the model, number of lagged variables and the number of regimes increase.

Instruction: Set $S = 1$, n and m for LRE models and set $S = 2$ or 3, n and m for MSRE models. Read the instruction of the code or the code explanation in the following section. We first show that the technique can be applied to LRE models as if no eigensystem were available. Then one can also infer how touch to apply this approach to MSRE models.

A. Gröbner basis approach for LRE models

For LRE models, the problem of computing all MSV solutions boils down to the system of polynomial equations implied by

$$A\Omega^2 - \Omega + B = 0_{n\times n}$$

Vectorizing this implies that there are n^2 quadratic equations in n^2 unknown ω_{ij}, which is the (i,j)-th element of Ω. When some of elements of B are zeros, the number of unknowns can be reduced. For simplicity, let m be the rank of B and construct arbitrary B such that the first $n-m$ columns are $n \times (n-m)$ matrix of zeros. The code gbmsre.m can also handle the LRE models by setting $P = 1$ and $S = 1$. Let $Model(n,m)$ denotes a LRE model with an n dimensional model and m being the rank of B. Then, the number of MSV solutions is $N = \frac{(n+m)!}{m!n!}$ as it is the same as the number of combination of choosing n out of $2n-(n-m) = n+m$. This can be derived from Generalized Schur Decomposition. Refer to qzmlre.m for detail.

The code GBmsre_Testing_computation_time.m computes the solution to a model specified by P, n and m using gbmsre.m. For an LRE model, set $P = 1$. The code generates $n \times n$ matrices A and B of random numbers such that the rank of B is m. Table 4 shows that the number of seconds required to compute the all solutions to $Model(n,m)$. The case of $m = 1$ is instructive because the number of $N = n+1$. The number of ideals implied by the system of the polynomial equations is equal to the total number of solutions. Gröbner basis approach is a way of finding these ideals, and therefore, we can infer how long it would take as the number of unknowns increases by one. Table 4 indicates that the computation time increases by a factor of around 2^N. This is the source of the problem why finding all solutions to a polynomial is difficult.

Table 4: **Computation Time for Gröbner basis in LRE Models**

n	m	N	Seconds	n	m	N	Seconds
1	1	2	< 1	2	2	6	< 1
2	1	3	< 1	3	2	10	< 1
3	1	4	< 1	4	2	15	≈ 1
9	1	10	≈ 1.5	5	2	21	≈ 7
10	1	11	≈ 3	6	2	28	≈ 548
11	1	12	≈ 4	7	2	36	$N.A$
12	1	13	≈ 12	3	3	20	≈ 3
13	1	14	≈ 35	4	3	35	≈ 4500
14	1	15	≈ 95	5	3	252	$N.A$
15	1	16	≈ 208	4	4	70	$N.A$

For $Model(n \geq 2, m = 2)$, the number of solutions explodes and it takes about 10 minutes for $Model(6,2)$ but computation fails to yield the solution as $n \geq 7$. The most general model that the Gröbner basis approach is successful is $Model(4, 3)$. $N.A$ in Table 4 is this case where gbmsre.m stops computation because of lack of memory within a day. Of course, when the structure of the model is simpler, the Gröbner basis technique may work for higher dimensional models, for instance, the rank of A is less than n, but this simulation exercise shows that it is extremely difficult to use this approach for non-trivial economic models.

B. Gröbner basis approach for MSRE models

One can guess that the problem is much more serious for MSRE models. Indeed, the model can be stated as $Model(n, m, S)$ where S is the number of regimes. There is no analytical form for N, the number of MSV solutions but $N \geq \Pi_{s=1}^{S} \frac{(n+m_s)!}{m_s!n!}$ in general. This is because when $P = I_S$, each regime collapses to $Model(n, m_s)$ Thus the minimum number of solution is the product of the number of solutions at each regime. For instance, $Model(n, n) = 2, 6, 20, ...$ for $n = m = 1, 2, 3...$ Therefore, $N \geq 4, 36, 400, ..$ for MSRE models $Model(n, n, S)$. This implies that the computation time will increase more than $2^4, 2^{36-4}, 2^{400-36}$. Indeed the method works extremely fast for $Model(1, 1, 2)$ with $N = 4$, and it takes several minutes for $Model(2, 2, 2)$ with $N = 44 > 36$. However, computation fails for $Model(3, 3, 2)$. Table 5 shows the MSRE models for which Gröbner basis works and does not work.

Table 5: **Computation Time for Gröbner basis in MSRE Models**

n	m	S	N	Seconds	n	m	S	N	Seconds
1	1	2	4	< 1	2	2	2	44	≈ 227
2	1	2	9	< 1	3	2	2	$N.A$	$N.A$
3	1	2	16	< 1	1	1	3	8	< 1
4	1	2	25	≈ 3	2	1	3	27	≈ 2
5	1	2	36	≈ 60	3	1	3	64	≈ 592
6	1	2	49	≈ 573	4	1	3	125	$N.A$
7	1	2	13	$N.A$	2	2	3	$N.A$	$N.A$

B Examples

B.1 Codes for Replicating Cho (2016)

All of the models turn out to be determinacy-admissible and the forward solution is now confirmed by the MOD method. Therefore, the results of Cho (2016) is necessary as well as sufficient.

B.1.1 The Fisherian Model

X1.m: This is a univariate MSRE model without predetermined variables considered by Davig and Leeper (2007) and Farmer et al. (2009):

$$\begin{aligned} \alpha(s_t)\pi_t &= E_t\pi_{t+1} + r_t, \\ r_t &= \rho r_{t-1} + \epsilon_t, \end{aligned}$$

where π_t and r_t are inflation and the real interest rate, respectively. When $\alpha(s_t) < (>)1$, monetary policy is passive (active). The model can be cast in the form of (12) as:

$$x_t = A(s_t)E_t x_{t+1} + C(s_t)z_t,\ z_t = \rho z_{t-1} + \epsilon_t,$$

where $x_t = \pi_t$, $z_t = r_t$, $A(s_t) = C(s_t) = 1/\alpha(s_t)$, $B(s_t) = 0$ and $R = \rho$. The forward solution will be of the form $x_t = \Gamma^*(s_t)z_t$ if it exists. Let $\rho = 0.9$, $p_{11} = 0.8$ and $p_{22} = 0.9$.

Case=1: This is an example of determinacy with $\alpha(1) = 0.95$ and $\alpha(2) = 1.5$. One may write the code as:

```
S=2; p11=0.8; p22=0.9; p12=1-p11; p21=1-p22; P=[p11 p12;p21 p22];
rho=0.9; alpha{1}=0.95; alpha{2}=1.5;
for i=1:S, A{i,1}=1/alpha{i}; C{i,1}=1/alpha{i}; end, R=rho;
[DET, FCC, OmegaK,GammaK,FK]=fmmsre(P,A,[],C,R);
```

The first three lines specify the number of regimes, transition probability matrix and A, C and R. The last line shows the function "fmmsre.m". Since no lagged variables are present, FCC(1) holds trivially because there is no forward iteration and so FCC1=1. DET(1) $= r(\bar{\Psi}_{\Omega^*\otimes\Omega^*}) = 0$ and DET(2) $= r(\Psi_{F^*\otimes F^*}) = 0.91 < 1$.

Since DET(1)$\times$DET(2)$= 0 < 1$, the model is determinacy-admissible and the model is determinate. Since FCC(2) $= r(\Psi_{R'\otimes F^*}) = 0.80 < 1$, the forward solution exists, which is given by $x_t = \Gamma^*(s_t)z_t$. GammaK{1,1} representing $\Gamma^*(1)$ is 6.11 and GammaK{2,1} =2.25.

Case=2: This is an example of indeterminacy with $\alpha(1) = 0.9$ and $\alpha(2) = 1.5$. Set alpha{1}=0.9 above and run the code. In this case, FCC(1)= 1. DET(1) = 0. DET(2)= $1.01 > 1$. The model is still determinacy-admissible, but indeterminate.

B.1.2 Regime-Switching Monetary Policy

X2.m: This is the model analyzed in Section 5 of Cho (2016). The model is given by:

$$\begin{aligned} \pi_t &= \beta E_t\pi_{t+1} + \kappa y_t + z_{S,t}, & (23)\\ y_t &= E_t y_{t+1} - \frac{1}{\sigma}(i_t - E_t\pi_{t+1}) + z_{D,t}, & (24)\\ i_t &= (1-\rho)\phi_\pi(s_t)\pi_t + \rho i_{t-1} + z_{MP,t}. & (25) \end{aligned}$$

In matrix form, $x_t = [\pi_t\ y_t\ i_t]'$, $z_t = [z_{S,t}\ z_{D,t}\ z_{MP,t}]'$, $\epsilon_t = [\epsilon_t^S\ \epsilon_t^D\ \epsilon_t^{MP}]'$, and,

$$B_1(s_t)x_t = A_1(s_t)E_t[x_{t+1}] + B_2 x_{t-1} + z_t,\ z_t = Rz_{t-1} + \epsilon_t,$$

$$B_1(s_t) = \begin{bmatrix} 1 & -\kappa & 0 \\ 0 & 1 & 1/\sigma \\ -(1-\rho)\phi_\pi(s_t) & 0 & 1 \end{bmatrix},\ A_1(s_t) = \begin{bmatrix} \beta & 0 & 0 \\ 1/\sigma & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \rho \end{bmatrix}.$$

The matrix R is now a matrix of zeros. Then the model is given by:

$$x_t = A(s_t)E_t[x_{t+1}] + B(s_t)x_{t-1} + C(s_t)z_t,$$

where $A(s_t) = B_1(s_t)^{-1}A_1(s_t)$, $B(s_t) = B_1(s_t)^{-1}B_2$ and $C(s_t) = B_1(s_t)^{-1}$. The parameter values are chosen as follows.

Common Parameters	$p_{11} = 0.85$, $p_{22} = 0.95$, $\beta = 0.99$, $\kappa = 0.132$, $\sigma = 1$, $\rho = 0.7$, $\rho_D = \rho_S = 0.8$, $\rho_{MP} = 0$.
Case=1: Determinate	$\phi_\pi(1) = 0.9$, $\phi_\pi(2) = 1.5$.
Case=2: Indeterminate	$\phi_\pi(1) = 0.8$, $\phi_\pi(2) = 1.5$.

The results are qualitatively similar to those of the Fisherian model. In the case of determinacy, FCC(1)=22. DET(1) $= r(\bar{\Psi}_{\Omega^*\otimes\Omega^*}) = 0.2839 < 1$ and DET(2) $= r(\Psi_{F^*\otimes F^*}) = 0.9539 < 1$. So the model is determinate and it is of course determinacy-admissible as DET(1)$\times$DET(2)$= 0.2708 < 1$. Since FCC(2) $= 0.7442 < 1$, the forward solution exists, which is given by $x_t = \Omega^*(s_t) + \Gamma^*(s_t)z_t$. In the case of indeterminacy, FCC(1)=24. DET(1) $= 0.2895 < 1$ and DET(2) $= 1.0082 > 1$. But the model is determinacy-admissible as DET(1)DET(2)$= 0.2919 < 1$.

X2figure.m: This code computes the locus of $\phi_\pi(1)$ and $\phi_\pi(2)$ such that $r(\Psi_{F^*\otimes F^*}) = 1$ by "fsolve.m", which partitions the parameter space into the determinacy and indeterminacy regions.

B.1.3 Regime-Switching Elasticity of Intertemporal Substitution

The code **X3.m** considers a New-Keynesian model with regime-switching elasticity of intertemporal substitution. The model is presented in Section 6.3 of the previous version of Cho (2016) "Characterizing Markov-Switching Rational Expectations (MSRE) models".

$$\pi_t = \beta E_t \pi_{t+1} + \kappa y_t + z_{S,t}, \tag{26}$$

$$y_t = E_t\left[\frac{\sigma(s_{t+1})}{\sigma(s_t)} y_{t+1}\right] - \frac{1}{\sigma(s_t)}(i_t - E_t\pi_{t+1}) + \frac{1}{\sigma(s_t)} z_{D,t}, \tag{27}$$

$$i_t = (1-\rho)\phi_\pi \pi_t + \rho i_{t-1} + z_{MP,t}. \tag{28}$$

This example differs from the Markov-switching monetary policy model in that A depends both on s_t and s_{t+1}. This new feature can be handled by simply specifying $A(i,j)$ for all $i,j = 1,2,...,S$. For a numerical exercise we have $\beta = 0.99$, $\kappa = 0.132$, $\phi_\pi = 1.5$, $\rho_D = \rho_S = 0.95$, $\rho_{MP} = 0$ and $\rho = 0.95$. For $\sigma(s_t)$ we set $\sigma(1) = 1$ and $\sigma(2)$. Since $\phi_\pi > 1$, the model would be determinate and this can be easily verified by $r(\bar{\Psi}_{\Omega^*\otimes\Omega^*}) = 0.56$ and $r(\Psi_{F^*\otimes F^*}) = 0.95$. Of course, the model is determinacy-admissible as the product of the two metrics is less than 1.

Remark The impulse response function is an optional output argument of the code "fmmsre.m". All three codes **X1.m, X2.m** and **X3.m** will display the impulse response functions starting at two different initial regimes.

B.2 Codes for Replicating Cho (2019)

1. ReplicatingFigure1.m produces Figure 1 of Cho (2019). It applies "fmlre.m" and "fmmsre.m" for more than 120,000 times of parameter combinations in about a couple of minutes. Just run this code.

2. Example_Corollary1_Fisher_FTPL.m is another example for Corollary 1 using the model with regime-switching in monetary and fiscal policies in the paper.

B.3 Sample Codes for Cho and Moreno (2019)

To be added.

C Modified Forward Method

The original forward method of Cho and Moreno (2011) for LRE models and the forward method Cho (2016) for MSRE models yields the forward solution different from the MOD solution *only when* the model has a special block-recursive structure. The modification of the original forward method aims at recovering the equivalence of the forward solution and the MOD solution for models with a block-recursive structure or a type of models that breaks down the equivalence of the two solutions for unknowns reasons for MSRE models. Refer to Cho (2019) for detail.

The information with which expectations are formed may not contain all of the state variables for block-recursive models. Thus let the original forward method use partial information. The information can be extended to include all of the state variables regardless of the model structure. This case uses full information. Modified forward method can be summarized as follows. First, apply the original forward method. If the forward solution is the MOD solution, that is, if the model is determinacy-admissible, then it is done. If not, use the forward method under full information. It suffices to illustrate the forward method for MSRE models because the forward method for LRE models is just a special case. The following briefly explains the original forward method and the forward method under full information.

C.1 Original Forward Method for MSRE models.

1. **Forward Representation of the Model (Proposition 3 of Cho (2016))**: There is a unique set of sequences $\Omega_k(s_t)$, $\Gamma_k(s_t)$ and $F_k(s_t, s_{t+1})$ such that

$$x_t = E_t[M_k(s_t, s_{t+1}, ..., s_{t+k})x_{t+k}] + \Omega_k(s_t)x_{t-1} + \Gamma_k(s_t)z_t, \tag{29}$$

where $\Omega_1(s_t) = B(s_t)$, $\Gamma_1(s_t) = C(s_t)$, $F_1(s_t, s_{t+1}) = A(s_t, s_{t+1})$ *for all* s_t and s_{t+1}, *and for* $k = 2, 3, ...,$

$$\begin{aligned}
\Omega_k(s_t) &= \{I_n - E_t[A(s_t, s_{t+1})\Omega_{k-1}(s_{t+1})]\}^{-1}B(s_t), && (30)\\
\Gamma_k(s_t) &= \{I_n - E_t[A(s_t, s_{t+1})\Omega_{k-1}(s_{t+1})]\}^{-1}C(s_t) && (31)\\
&\quad + E_t[F_k(s_t, s_{t+1})\Gamma_{k-1}(s_{t+1})R(s_{t+1})], \\
F_k(s_t, s_{t+1}) &= \{I_n - E_t[A(s_t, s_{t+1})\Omega_{k-1}(s_{t+1})]\}^{-1}A(s_t, s_{t+1}). && (32)
\end{aligned}$$

- **No-bubble Condition (NBC)** holds if $\lim_{k\to\infty} E_t[M_k(s_t, s_{t+1}, ..., s_{t+k})x_{t+k}] = 0_{n\times 1}$ when expectations are formed with that solution.

- **Forward Convergence Condition (FCC)**

Table 6: **Forward Convergence Conditions**

Notations	Concepts	How to check
FCC1	$\Omega^*(s_t) = \lim_{k\to\infty} \Omega_k(s_t)$	every element of $\Omega_k(s_t) - \Omega_{k-1}(s_t) \to 0$
FCC2	$\Gamma^*(s_t) = \lim_{k\to\infty} \Gamma_k(s_t)$	$r(\Psi_{R'\otimes F^*}) < 1$

- The forward solution is:

$$x_t = \Omega^*(s_t)x_{t-1} + \Gamma^*(s_t)z_t. \tag{33}$$

  - **Remark 1.** In practice, $\Omega^*(s_t)$ always exists except for the case in which the model has no real-valued solutions – the case the FCC should not hold.
  - **Remark 2.** $F^*(s_t, s_{t+1}) = \lim\limits_{k\to\infty} F_k(s_t, s_{t+1})$ exists whenever $\Omega^*(s_t)$ exists.
  - **Remark 3.** Existence of $\Gamma^*(s_t)$ can be checked easily by computing $r(\Psi_{R'\otimes F^*})$. One can understand why this is true from (31). That is, $\Gamma_k(s_t)$ explodes if $r(\Psi_{R'\otimes F_K}) > 1$.

2. **Property of the Forward Solution (Proposition 4 of Cho (2016)):** The forward solution is the unique RE solution that satisfies the NBC.

3. **Classification of the solutions by the forward method (Proposition 5 and 6 of Cho (2016)):** Suppose that the forward solution $\Omega^*(s_t)$ exists and $r(\bar{\Psi}_{\Omega^*\otimes\Omega^*})r(\Psi_{F^*\otimes F^*}) < 1$. Then $\Omega^*(s_t) = \Omega^{MOD}(s_t)$. Classify the model following Table 3 (See Cho (2019)). Done.

C.2 Forward Method for MSRE models under Full Information.

If $r(\bar{\Psi}_{\Omega^*\otimes\Omega^*})r(\Psi_{F^*\otimes F^*}) \geq 1$, forward one period ahead and take expectations as:

$$E_t[x_{t+1}] = E_t[k_{t+1}] + E_t[B(s_{t+1})]x_t, \tag{34a}$$

which must be true under rational expectations where $k_t = E_t[A(s_t, s_{t+1})x_{t+1}]$. $E_t[B(s_{t+1})]$ at each regime s_t can be easily computed by $\sum_j^S P(i,j)B(s_{t+1} = j)$ at each $s(t) = i = 1, ..., S$. Then the model is transformed as follows.

$$\begin{aligned} x_t &= k_t + B(s_t)x_{t-1} && \text{(35a)} \\ k_t &= E_t[A(s_t, s_{t+1})x_{t+1}] + H(E_t[x_{t+1}] - E_t[k_{t+1}] - E_t[B(s_{t+1})]x_t), && \text{(35b)} \end{aligned}$$

where H is an $n \times n$ matrix in which every single element is arbitrary but non-zero. Applying the expectational relation of the whole model is innocuous because it must hold regardless of block-recursiveness of the model. By doing this, the model is no-longer block-recursive and has no autonomous block. Now we write this into the original form of the model in the following way. Let $y_t = [x_t' \ k_t']'$ be a $2n \times 1$ vector. Notice that $E_t[B(s_{t+1})]$ is a function of s_t. Collecting the coefficient matrices yields:

$$\begin{bmatrix} I_n & -I_n \\ HE_t[B(s_{t+1})] & I_n \end{bmatrix} y_t = E_t\left[\begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ A(s_t, s_{t+1}) + H & -H \end{bmatrix} y_{t+1}\right] + \begin{bmatrix} B(s_t) & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix} y_{t-1},$$

Finally, by multiplying the inverse of the coefficient matrix of y_t, we have the following form of the model:

$$y_t = E_t[A^y(s_t, s_{t+1})y_{t+1}] + B^y(s_t)y_{t-1}. \tag{36}$$

Forward method under full information is to apply the forward method to (36) and obtain the solution for y_t.

$$y_t = (\Omega^y(s_t))^* y_{t-1}. \tag{37}$$

The solution $(\Omega^y(s_t))^*$ is $2n \times 2n$ and since x_t is the first $n \times 1$ subvector of y_t, the first $n \times n$ component of $(\Omega^y(s_t))^*$ is the forward solution of the original model under the modified forward method.

C.3 Existence of the Forward Solution

Both the original model (12) and the adjusted model (36) has the same form. Therefore, we can examine the existence condition for the forward solution using the same formula as (30) and (32). Under the modified forward method, $\Omega_k(s_t)$ can be interpreted as $\Omega_k^y(s_t)$ in (37). The condition for the existence of the forward solution can be understood by differentiating and vectoring (30) such that:

$$\begin{bmatrix} vec(d\Omega_k(1)) \\ ... \\ vec(d\Omega_k(S)) \end{bmatrix} = \begin{bmatrix} \Psi_{\Omega'_{k-1}\otimes F_{k-1}} \end{bmatrix} \begin{bmatrix} vec(d\Omega_{k-1}(1)) \\ ... \\ vec(d\Omega_{k-1}(S)) \end{bmatrix}. \tag{38}$$

Proposition 4 of Cho (2019) formally states the convergence condition for Ω_k. There are two possibilities:

1. $r(\Psi_{\Omega'_{k-1}\otimes F_{k-1}}) < 1$ for all $k > 1$.

2. $r(\Psi_{\Omega'_{k-1}\otimes F_{k-1}}) \geq 1$ for some $k > 1$ and $u'_k vec(d\Omega_{k-1}) = 0_{n^2S\times 1}$ for every eigenvector u_k associated with an unstable root of $\Psi_{\Omega'_{k-1}\otimes F_{k-1}}$.

Under the original forward method, Proposition 4 states that the equivalence of the forward solution and the MOD solution may break down only in the second case. In that case, even if $\Psi_{\Omega'_{k-1}\otimes F_{k-1}}$ contains a root larger than unity, it does not affect Ω_k, thus the forward solution can still exist. The forward method under full information is to adjust the model such that Case 2 of Proposition does not arise in the class of determinacy-admissible models.[2] Then Case 1 implies that in the MSRE framework, the forward solution exists for a broader class of models including determinacy-inadmissible models because, $r(\Psi_{\Omega^{*\prime}\otimes F^*}) < 1$ can be consistent with $r(\bar{\Psi}_{\Omega^*\otimes\Omega^*})r(\Psi_{F^*\otimes F^*}) \geq 1$. That is, if $r(\bar{\Psi}_{\Omega^*\otimes\Omega^*})r(\Psi_{F^*\otimes F^*}) < 1$, then $r(\Psi_{\Omega^{*\prime}\otimes F^*}) < 1$, but the converse is not true in general. An extensive experiment so far has never found a single case in which the forward solution is not the MOD solution in atheoretical and economic examples. Nevertheless, the equivalence to a model with a real-valued MOD solution is an open question to be explored in the future.

[2]When a model has completely decoupled equations, Case 2 still exists even under the forward method under full information. The example in Section 4 when $c = 0$ is such a case. However, recall that in that case, the solution with the smallest generalized eigenvalues does not exists, thus the model is determinacy-inadmissible. Moreover, the forward solution coincides with the MOD solution.

References

Cho, Seonghoon, "Sufficient conditions for determinacy in a class of Markov-switching rational expectations models," *Review of Economic Dynamics*, 2016, *21*, 182–200.

_ , "Determinacy and Classification of Markov-Switching Rational Expectations Models," *Available at SSRN 3229215*, 2019.

_ **and Antonio Moreno**, "The Forward Method as a Solution Refinement in Rational Expectations Models," *Journal of Economic Dynamics and Control*, 2011, *35* (3), 257–272.

_ **and** _ , "Has Fiscal Policy Saved the Great Recession?," *Working Paper*, 2019.

_ **and Bennett T McCallum**, "Refining linear rational expectations models and equilibria," *Journal of Macroeconomics*, 2015, *46*, 160–169.

Davig, Troy and Eric M. Leeper, "Generalizing the Taylor Principle," *American Economic Review*, 2007, *97* (3), 607–635.

Decker, Wolfram, Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann, "Singular 4-1-2 — A computer algebra system for polynomial computations," 2019.

Farmer, Roger E.A., Daniel F. Waggoner, and Tao Zha, "Understanding Markov-Switching Rational Expectations Models," *Journal of Economic Theory*, 2009, *144* (5), 1849–1867.

Foerster, Andrew, Juan F Rubio-Ramírez, Daniel F Waggoner, and Tao Zha, "Perturbation methods for Markov-switching dynamic stochastic general equilibrium models," *Quantitative Economics*, 2016, *7* (2), 637–669.

Sims, Christopher A., "Solving Linear Rational Expectations Models," *Computational Economics*, 2002, *20* (1), 1–20.

Sims, Christopher A, "On the genericity of the winding number criterion for linear rational expectations models," Technical Report, Citeseer 2007.