

CAMP Working Paper Series
No 10/2014

Efficient Perturbation Methods for Solving Regime-Switching DSGE Models

Junior Maih



© Authors 2014.

This paper can be downloaded without charge from the CAMP website <http://www.bi.no/camp>

Efficient Perturbation Methods for Solving Regime-Switching DSGE Models*

Junior Maih[†]

January 16, 2015

Abstract

In an environment where economic structures break, variances change, distributions shift, conventional policies weaken and past events tend to reoccur, economic agents have to form expectations over different regimes. This makes the regime-switching dynamic stochastic general equilibrium (RS-DSGE) model the natural framework for analyzing the dynamics of macroeconomic variables. We present efficient solution methods for solving this class of models, allowing for the transition probabilities to be endogenous and for agents to react to anticipated events. The solution algorithms derived use a perturbation strategy which, unlike what has been proposed in the literature, does not rely on the partitioning of the switching parameters. These algorithms are all implemented in RISE, a flexible object-oriented toolbox that can easily integrate alternative solution methods. We show that our algorithms replicate various examples found in the literature. Among those is a switching RBC model for which we present a third-order perturbation solution.

JEL Classification: C6, E3, G1.

Keywords: DSGE, Markov switching, Sylvester equation, Newton algorithm, perturbation, matrix polynomial

*First version: August 1, 2011. This Working Paper should not be reported as representing the views of Norges Bank. The views expressed are those of the author and do not necessarily reflect those of Norges Bank. Some of the ideas in this paper have often been presented under the title : “Rationality in Switching Environments”. I thank Hilde Bjørnland, Oren Levintal, Dan Waggoner, Tao Zha, and seminar participants at the Atlanta Fed (Sept. 2011), IMF, Norges Bank, JRC, the National Bank of Belgium, dynare conferences (Zurich, Shanghai), ECB, Board of Governors of the Federal Reserve System and the 87th WEAI conference for comments.

[†]Norges Bank and Centre for Applied Macro and Petroleum economics, BI Norwegian Business School. Email: junior.maih@norges-bank.no

1 Introduction

In an environment where economic structures break, variances change, distributions shift, conventional policies weaken and past events (e.g. crises) tend to reoccur, economic agents have to form expectations over different regimes.¹ This makes the regime-switching DSGE (RS-DSGE) model the natural framework for analyzing the dynamics of macroeconomic variables.² This class of models is especially important because many policy questions of interest seem to be best answered/addressed in a framework of changing parameters or, more generally, changing regimes. Some of those questions are:

- what actions should we undertake today given the non-zero likelihood of a bad state occurring in the future?
- what can we expect of the dynamics of the macro-variables we care about when policy is constrained?
- how is the economy stabilized when policy is constrained?

Not surprisingly then, besides the ever-growing empirical literature using RS-DSGE models, many efforts have been directed towards solving those models. In that respect, the literature has considered three main angles of attack. One strand of the literature considers approximating the solution of those models using “global” methods. Examples include [Davig et al. \(2011\)](#), [Bi and Traum \(2013\)](#) and [Richter et al. \(2014\)](#). Just as in constant-parameter DSGE models, global approximation methods in RS-DSGE models face problems of curse of dimensionality, reliance on a pre-specified set of grid points typically constructed around one steady state although the model may have many, etc. The curse of dimensionality in particular, implies that the number of state variables has to be as small as possible and even solving small models involves substantial computational costs.

¹Changes in shock variances have been documented by [Stock and Watson \(2003\)](#), [Sims and Zha \(2006\)](#), [Justiniano and Primiceri \(2008\)](#), while breaks in structural parameters have been advocated by [Bernanke et al. \(1999\)](#), [Lubik and Schorfheide \(2004\)](#), [Davig and Leeper \(2007\)](#). Other papers have also documented changes in both variances and structural parameters. Examples include [Smets and Wouters \(2007\)](#), [Svensson and Williams \(2007, 2009\)](#) and [Cogley et al. \(2012\)](#).

²To the extent that these features are present in the data, constant-parameter models could be misleading in their predictions and their implications for policymaking. Another popular way of modeling switches in parameters is the MSVAR model. This model is conditionally linear and does not have forward-looking terms. We, therefore, view it as a special case of the more general RS-DSGE framework.

A second group of techniques applies Markov switching to the parameters of linear or linearized DSGE models. Papers in this class include [Farmer et al. \(2011\)](#), [Cho \(2014\)](#), [Svensson and Williams \(2007\)](#) to name a few. One advantage of this approach is that one can handle substantially larger problems than the ones solved using global methods. Insofar as the starting point is a linear model, all one has to worry about is how to compute and characterize the solution of the model. If the original model is nonlinear, however, a first-order approximation may not be sufficient to approximate the nonlinear dynamics implied by the true policy functions of the model. Because agents are aware of the switching process, the nonlinearity of the original model also implies that one cannot assume away the switches in the parameters for the sake of linearization and then reapply them to the model parameters once the linearized form is obtained. This is especially important because switching parameters may imply different steady states or attractors. Therefore, consistently linearizing the model while taking into account the switching parameters calls for a different strategy.

Finally, the third group of techniques attempts to circumvent or find a work-around to the problems posed by the first two groups. More specifically, this literature embeds switching in perturbation solutions whose accuracy can be improved with higher and higher orders.³ This is the approach followed by [Barthelemy and Marx \(2011\)](#), [Foerster et al. \(2013, 2014\)](#) and in this paper.

For many years we have been developing, in the RISE toolbox, algorithms for solving and estimating models of switching parameters including DSGEs, VARs, SVARs, optimal policy models of commitment, discretion and loose commitment.⁴ In the present paper, however, we focus on describing the theory behind the routines implementing the RS-DSGE model in RISE.⁵

The approach in RISE is more general than the ones discussed earlier or found in the literature. In contrast to [Foerster et al. \(2013, 2014\)](#) and the papers cited above, in our derivation of higher-order perturbations we allow for endogenous transition probabilities.⁶ We also allow for anticipated events or shocks following [Maih \(2010\)](#) and [Juillard and Maih \(2010\)](#). This feature is useful as it offers an alternative to the news shocks strategy that has often been used to analyze e.g. the effects of forward guidance (e.g. [Campbell et al. \(2012\)](#), [Del Negro et al. \(2013\)](#), [Gavin et al. \(2014\)](#)).

³Model accuracy is often measured by the Euler approximation errors. It is important to note that a high accuracy is not synonymous with the computed policy functions being close to the true ones

⁴RISE is a matlab-based object-oriented toolbox. It is available, free of charge, at https://github.com/jmaih/RISE_toolbox.

⁵We plan to describe algorithms for other modules of RISE in subsequent papers.

⁶[Barthelemy and Marx \(2011\)](#) also consider endogenous probabilities but only in forward-looking models for which they establish the existence of a unique stable equilibrium.

The introduction of an arbitrary number of anticipated shocks does come at a cost though: the number of cross terms to keep track of increases rapidly with the order of approximation and computing the coefficients on those terms beyond the first order becomes cumbersome. This problem is circumvented by exploiting a simple trick by [Levintal \(2014\)](#), who shows a way of computing higher-order approximations by keeping all state variables in a single block.⁷

But before computing higher-order perturbations, two problems relative to the choice of the approximation point and the solving of a quadratic matrix equation, have to be addressed. With respect to the choice of the approximation point, [Foerster et al. \(2014\)](#) propose a “partition perturbation” strategy in which the switching parameters are separated into two groups: one group comprises the switching parameters that affect the (unique) steady state and another group collects the switching parameters that do not affect the steady state. Here also, our approach as implemented in RISE is more general, more flexible yet simpler: it does not require any partitioning of the switching parameters and is therefore more efficient. Moreover, it allows for the possibility of multiple steady states and delivers the results of the “partition perturbation” of [Foerster et al. \(2014\)](#) as a special case.

When it comes to solving the system of quadratic matrix equations implied by the first-order perturbation, [Foerster et al. \(2013, 2014\)](#)’s proposal is to use the theory of Gröbner bases (see [Buchberger \(1965, 2006\)](#)) to find all the solutions of the system and then apply the engineering concept of mean square stability (MSS) to each of the solutions as a way to check whether the Markov-Switching DSGE (MS-DSGE)⁸ system admits a single stable solution. While the benefits of a procedure that can generate all possible solutions of a system of polynomial equations are undeniable, we argue that such an approach may not be practical or suitable for systems of the size that we have been accustomed to in policy institutions: both the computation of Gröbner bases and the checking of determinacy by MSS are prohibitively expensive operations in medium-to-large scale models.⁹ The [Foerster et al. \(2013,](#)

⁷Before the new implementation, the higher-order perturbation routines of RISE separated the state variables into three blocks: endogenous variables, perturbation parameter, exogenous shocks. Cross products of those had to be taken explicitly, calculated and stored separately.

⁸We use MS-DSGE to denote RS-DSGE models in which transition probabilities are constant.

⁹In the branch of mathematics called Algebraic Geometry, solving systems of polynomial equations is a very active topic and Gröbner bases are just one of the plethora of techniques available. There various other techniques based on “resultants” (see e.g. [Canny and Emiris \(1993\)](#), [Emiris \(1996\)](#), [Emiris and Mourrain \(1999\)](#), [Dreesen et al. \(2012\)](#)), on “homotopy” (e.g. [Garcia and Zangwill \(1979\)](#), [Morgan and Sommese \(1987\)](#)) as well as the “Ritt-Wu” methods (e.g. [Ritt \(2008\)](#), [Wu \(1978\)](#)). All those techniques have their own advantages and drawbacks but are all known to be very computationally intensive and typically work well on small systems. Gröbner bases in

2014) approach has two further limitations: (1) stability of first-order approximation does not imply stability of higher-order approximations, even in constant-parameter DSGE models; (2) there is no stability concept for switching models with endogenous transition probabilities.

Because we are ultimately interested in estimating those models in order to make them really useful for policymaking, we take a more modest route: we derive efficient functional iterations and Newton algorithms that are suitable for solving relatively large systems. As an example, we have successfully solved a second-order perturbation of a model of 272 equations using our algorithms. We further demonstrate the efficiency and usefulness of our solution methods by applying them to various examples found in the literature. Our algorithms easily replicate the results found by other authors. Among the examples we consider is a switching model by Foerster et al. (2014) for which we present a third-order perturbation solution.

The rest of the paper proceeds as follows. Section 2 introduces the notation we use alongside the generic regime-switching model that we aim to solve. Then Section 3 derives the higher-order approximations to the solution of the model. At the zeroth order, we present various flexible strategies for choosing the approximation point. Section 4 provides some details about how we can solve a generalized system of coupled Sylvester equations. These equations arise both in higher-order approximations and in one of our Newton algorithms for solving the first-order approximation. Building on the previous section, Section 5 takes on the solving of the quadratic matrix polynomial arising in the first-order approximation. Three iterative solution techniques are proposed: one functional iteration and two Newton algorithms. Section 6 evaluates the performance of the proposed algorithms and Section 7 concludes.

2 The regime-switching DSGE model

2.1 The economic environment

We are interested in characterizing an environment in which parameters (and potentially equations) switch in a model that is potentially already nonlinear even in the absence of switching parameters. In that environment we would like to allow for the transitions controlling parameter switches to be endogenous and not just exogenous as customarily found in the literature. Finally, recognizing that at the time of making decisions agents may have information about future events, it is desirable that

particular are known to be an expspace-complete (or expspace-hard) problem i.e. a large amount of space is required for storing intermediate results.

future events – forward guidance on the behavior of policy is an example of such a possibility – which may or many not materialize in the end, influence the current behavior of private agents.

2.2 Dating and notation conventions

The dating convention used in this paper is different from the widely used convention of [Schmitt-Grohe and Uribe \(2004\)](#) in which the dating of the variables refers to the beginning of the period. Instead we rely on the also widely used dating convention in which the dating of the variables refers to the end of the period. In that way, the dating determines the period in which the variable of interest is known as opposed to the period in which it is used.¹⁰ This is the type of notation used for instance in [Adjemian et al. \(2011\)](#).

Some solution methods for constant-parameter DSGE models (e.g. [Klein \(2000\)](#), [Sims \(2002\)](#), [Schmitt-Grohe and Uribe \(2004\)](#)) stack variables of different periods. This type of notation has also been used in the context of MS-DSGE models by [Farmer et al. \(2011\)](#). This notation is not appropriate for the type of problems this paper aims to solve. In addition to forcing the creation of auxiliary variables and thereby increasing the size of the system, it also makes it cumbersome to compute expectations of variables in our context since future variables could belong to a state that is different from the current one. Clearly, doing so may restrict the class of problems one can solve or give the wrong answer in some types of problems.¹¹ The stacking of different time periods therefore is not appealing for our purposes.

2.3 The generic model

Many solution approaches, like [Farmer et al. \(2011\)](#), [Svensson and Williams \(2007\)](#) or [Cho \(2014\)](#), start out with a linear model and then apply a Markov switching to the parameters. This strategy is reasonable as long as one takes a linear specification as the structural model. When the underlying structural model is nonlinear, however, the agents are aware of the nonlinear nature of the system and of the switching process. This has implications for the solutions based on approximation and for the decision rules. For instance, an important result by [Foerster et al. \(2013\)](#) is that the

¹⁰For instance, in the notation used in this paper we would write a capital accumulation equation as $K_t = (1 - \delta) K_{t-1} + I_t$ rather than $K_{t+1} = (1 - \delta) K_t + I_t$.

¹¹In the [Farmer et al. \(2011\)](#) procedure for instance, the Newton algorithm is constructed on the assumption that the coefficient matrix on forward-looking variables depends only on the current regime.

first-order perturbation may be non-certainty equivalent. Furthermore, switching parameters may imply different steady states or attractors.¹² Hence starting out with a linear specification may miss these important points.

The problem to solve is

$$E_t \sum_{r_{t+1}=1}^h \pi_{r_t, r_{t+1}}(\mathcal{I}_t) \tilde{d}_{r_t}(v) = 0$$

where E_t is the expectation operator, $\tilde{d}_{r_t} : \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_d}$ is a $n_d \times 1$ vector of possibly nonlinear functions of their argument v (defined below), $r_t = 1, 2, \dots, h$ is the regime a time t , $\pi_{r_t, r_{t+1}}(\mathcal{I}_t)$ is the transition probability for going from regime r_t in the current period to regime r_{t+1} in the next period. This probability is potentially endogenous in the sense that it is a function of \mathcal{I}_t , the information set at time t . The only restriction imposed on the endogenous switching probabilities is that the parameters affecting them do not switch over time and that the variables entering those probabilities have a unique steady state.¹³

We refer to this model as the RS-DSGE model. Its special case in which the transition probabilities are constant will be referred to as Markov-Switching DSGE (MS-DSGE) model.

The $n_v \times 1$ vector v is defined as

$$v \equiv \left[b_{t+1}(r_{t+1})' \quad f_{t+1}(r_{t+1})' \quad s_t(r_t)' \quad p_t(r_t)' \quad b_t(r_t)' \quad f_t(r_t)' \quad p'_{t-1} \quad b'_{t-1} \quad \varepsilon'_t \quad \theta'_{r_{t+1}} \right]' \quad (1)$$

where :

- s_t is a $n_s \times 1$ vector of static variables. Those are the variables appearing in the model only at time t .
- f_t is a $n_f \times 1$ vector of forward-looking variables. Those are variables appearing in the model both at time t and at time $t + 1$.
- p_t is a $n_p \times 1$ vector of predetermined variables. Those variables appear in the model at time t and at time $t - 1$.
- b_t is a $n_b \times 1$ vector of “both” variables. Those are variables that are both predetermined and forward-looking

¹²[Aruoba et al. \(2014\)](#) give an example of a system that can exhibit both a targeted-inflation steady state and a deflationary steady state.

¹³RISE automatically checks for these requirements.

- ε_t is a $n_\varepsilon \times 1$ vector of shocks with $\varepsilon_t \sim N(0, I_{n_\varepsilon})$
- $\theta_{r_{t+1}}$ is a $n_\theta \times 1$ vector of switching parameters appearing with a lead in the model.

This classification of the variables is done automatically by the parser in RISE and so, all the user has to do is to provide a model file containing the equations of the model written in the most natural way. Note that we do not declare the parameters of the current regime r_t . They are implicitly attached to \tilde{d}_{r_t} , which also reflects that the model equations can switch and not just the parameters. Also note that we could get rid of the parameters of future regimes ($\theta_{r_{t+1}}$) by declaring auxiliary variables, as we will formally show in section (3.2). The resulting auxiliary variables are forward-looking.

If we define the $n_d \times 1$ vector $d_{r_t, r_{t+1}}$ as $d_{r_t, r_{t+1}} \equiv \pi_{r_t, r_{t+1}}(\mathcal{I}_t) \tilde{d}_{r_t}$, the objective becomes

$$E_t \sum_{r_{t+1}=1}^h d_{r_t, r_{t+1}}(v) = 0 \quad (2)$$

2.3.1 State variables

Following [Maih \(2010\)](#) and [Juillard and Maih \(2010\)](#), we assume that the agents have information for all or some of the shocks $k \geq 0$ periods ahead into the future. And so, including a perturbation parameter σ , we define an $n_z \times 1$ vector of state variables as

$$z_t \equiv [p'_{t-1} \quad b'_{t-1} \quad \sigma \quad \varepsilon'_t \quad \varepsilon'_{t+1} \quad \cdots \quad \varepsilon'_{t+k}]'$$

where $n_z = n_p + n_b + (k + 1) n_\varepsilon + 1$.

This strategy of modeling “anticipated events” or “anticipated shocks” differs from the alternative approach commonly referred to as “news shocks” in many ways. First, solving the model with anticipated shocks does not require a modification of the equations of the original system, in contrast to “news shocks” that are typically implemented by augmenting the law of motion of a shock process with additional shocks. Secondly, in the anticipated shocks approach, future events are discounted while in the news shocks approach the impact of a news shock does not depend on the horizon at which the news occurs.¹⁴ Thirdly, an anticipated shock is genuinely

¹⁴Discounting makes it possible to analyze the effects of “hard conditions” as well as “soft conditions” on the future information: the discounting of future events depends on the uncertainty around those events.

a particular structural shock in the original system, while in the news shocks, it is a different iid shock with no other interpretation than a “news” and unrelated to any structural shock in the system. Because it is unrelated, it will have its own distribution independently of other parts of the system. Fourthly, the estimation of models of news shocks requires additional variables to be declared as observables and enter the measurement equation. The estimation procedure then tries to fit the future information in the same way it fits the other observable variables. This feature makes Bayesian model comparison infeasible since the comparison of two models requires that they have the same observable variables. In contrast, in the estimation of models with anticipated shocks, the anticipated information, which may not materialize is separated from the actual data. Model comparison remains possible since the anticipated information never enters the measurement equations. Finally, in the anticipated shocks approach the policy functions are explicitly expressed in terms of leads of future shocks as opposed to lags in the news shocks approach.

2.3.2 General solution

Denoting by $y_t(r_t)$, the $n_y \times 1$ vector of all the endogenous variables, where $n_y = n_s + n_p + n_b + n_f$, we are interested in solutions of the form

$$y_t(r_t) \equiv \begin{bmatrix} s_t(r_t) \\ p_t(r_t) \\ b_t(r_t) \\ f_t(r_t) \end{bmatrix} = \mathcal{T}^{r_t}(z_t) \equiv \begin{bmatrix} \mathcal{S}^{r_t}(z_t) \\ \mathcal{P}^{r_t}(z_t) \\ \mathcal{B}^{r_t}(z_t) \\ \mathcal{F}^{r_t}(z_t) \end{bmatrix} \quad (3)$$

In general, there is no analytical solution to (2) even in cases where \tilde{d}_{r_t} or $d_{r_t, r_{t+1}}$ is linear. In this paper we rely on a perturbation that will allow us to approximate the decision rules in (3). We can then solve these approximated decision rules by inserting their functional forms into (2) and its derivatives. This paper develops methods for doing that.

3 Approximations

For the subsequent derivations, it is useful to define for all $g \in \{s, p, b, f\}$, an $n_g \times n_y$ matrix λ_g that select the solution of g -type variables in \mathcal{T} or y . We also define $\lambda_x \equiv \begin{bmatrix} \lambda_p \\ \lambda_b \end{bmatrix}$ and $\lambda_{bf} \equiv \begin{bmatrix} \lambda_b \\ \lambda_f \end{bmatrix}$ as the selector for p-b and b-f variables respectively. In the same way, we define for all $g \in \{p_{t-1}, b_{t-1}, \sigma, \varepsilon_t, \varepsilon_{t+1}, \dots, \varepsilon_{t+k}\}$, a matrix m_g of size $n_g \times n_z$ that selects the g -type variables in the state vector z_t .

Since the solution is in terms of the vector of state variables z_t , we proceed to expressing all the variables in the system as a function of z_t . Since both $b_{t+1}(r_{t+1})$ and $f_{t+1}(r_{t+1})$ appear in the system (1) and given the solution (3) we need to express z_{t+1} as a function of z_t as well. This is given by $z_{t+1} = h^{r_t}(z_t) + uz_t$, where

$$h^{r_t}(z_t) \equiv [(\lambda_x \mathcal{T}^{r_t}(z_t))' \quad (m_\sigma z_t)' \quad (m_{\varepsilon,1} z_t)' \quad \cdots \quad (m_{\varepsilon,k} z_t)' \quad (0_{n_\varepsilon \times 1})']'$$

and u is a $n_z \times n_z$ random matrix defined by

$$u \equiv \begin{bmatrix} 0_{(n_p+n_b+1+kn_\varepsilon) \times n_z} \\ \varepsilon_{t+k+1} m_\sigma \end{bmatrix} \quad (4)$$

The most natural way to handle the presence of future switching parameters in the system of the current regime is to define auxiliary variables. Doing so makes it easy to handle cases where the parameters of future regimes appear in the form of $\theta_{r_{t+\tau}}$ with $\tau > 1$. For the case where only next-period parameters appear, i.e. $\tau = 1$, one can take a shortcut like Foerster et al. (2013, 2014), and postulate a perturbation solution for $\theta_{r_{t+1}}$ as.¹⁵

$$\theta_{r_{t+1}} = \bar{\theta}_{r_t} + \hat{\sigma} \hat{\theta}_{r_{t+1}} \quad (5)$$

In this respect, this paper differs from Foerster et al. (2013, 2014) in two important ways. First, $\bar{\theta}_{r_t}$ need not be the ergodic mean of the parameters as will be discussed below. Secondly, conditional on being in regime r_t , perturbation is never done with respect to the θ_{r_t} parameters. Perturbation is done only with respect to the parameters of the future regimes ($\theta_{r_{t+1}}$) that appear in the system for the current regime (r_t).

Given the solution, we can now express vector v in terms of the state variables

$$v = \begin{pmatrix} \lambda_{bf} \mathcal{T}^{r_{t+1}} (h^{r_t}(z_t) + uz_t) \\ \mathcal{T}^{r_t}(z_t) \\ m_p z_t \\ m_b z_t \\ m_{\varepsilon,0} z_t \\ \bar{\theta}_{r_t} + \hat{\theta}_{r_{t+1}} m_\sigma z_t \end{pmatrix} \quad (6)$$

and the objective function (2) becomes

$$E_t \sum_{r_{t+1}=1}^h d_{r_t, r_{t+1}}(v(z_t, u)) = 0 \quad (7)$$

¹⁵We formally show in Section (3.2) that this expression can be derived as a first-order approximation of an endogenous auxiliary variable.

Having expressed the problem to solve in terms of state variables consolidated in a single vector z_t , as in [Levintal \(2014\)](#), we stand ready to take successive Taylor approximations of (7) to find the perturbation solutions. This strategy of consolidating all state variables in one block is particularly useful when it comes to computing higher-order cross derivatives. By not separating state variables, we always have one block of cross products no matter the order of approximation instead of an exponentially increasing number of cross blocks.

3.1 Zero-th order perturbation

The first step of the perturbation technique requires the choice of the approximation point. In a constant-parameter world, we typically approximate the system around the steady state, the resting point to which the system will converge in the absence of future shocks. In a switching environment the choice is not so obvious any more.

Approximation around the ergodic mean¹⁶ [Foerster et al. \(2013, 2014\)](#) propose to take a perturbation of the system around its ergodic mean. This ergodic mean can be found by solving $\tilde{d}_{\bar{r}}(b_t, f_t, s_t, p_t, b_t, f_t, p_t, b_t, 0, \bar{\theta}) = 0$. We use the notation $\tilde{d}_{\bar{r}}$ to indicate that the parameters of the current regime r_t have been replaced by those of the ergodic mean and $\bar{\theta}$ is the ergodic mean of the future switching parameters. The ergodic mean, however, need not be an attractor or a resting point, a point towards which the system will converge in the absence of further shocks. We propose two further possibilities.

Regime-specific steady states¹⁷ The first one is to approximate the system around regime-specific means. The system may not be stable at the mean in a certain regime, but at least we assume that if the system happens to be exactly at one of its regime-specific means, it will stay there in the absence of any further shocks. We compute those means by solving

$$\tilde{d}_{r_t}(b_t(r_t), f_t(r_t), s_t(r_t), p_t(r_t), b_t(r_t), f_t(r_t), p_t(r_t), b_t(r_t), 0, \theta_{r_t}) = 0$$

The intuition behind this strategy is two-fold. On the one hand, it is not too difficult to imagine that the relevant issue for rational agents living in a particular state of the system at some point in time is to insure against the possibility of

¹⁶In the RISE toolbox, this is triggered by the option “unique”.

¹⁷This approach is the default behavior in the RISE toolbox.

switching to a different state, *and not to the ergodic mean*. On the other hand, from a practical point of view, the point to which the system is to return matters for forecasting. Many inflation-targeting countries have moved from a regime of high inflation to a regime of lower inflation. Approximating the system around the ergodic mean in this case implies that the unconditional forecasts will be pulled towards a level that is consistently higher than the recent history of inflation, which is likely to yield substantial forecast errors. All this contributes to reinforcing the idea that the ergodic mean is not necessarily an attractor.

Approximation around an arbitrary point¹⁸ In the second possibility, one can impose an arbitrary approximation point. If the point of approximation is chosen arbitrarily, obviously, none of the two equations above will hold and a correction will be needed in the dynamic system, with consequences for the solution as well. This approach may be particularly useful in certain applications, e.g. a situation in which one of the regimes persistently deviates from the reference steady state for an extended period of time. The approach also bears some similarity with the constant-parameter case where the approximation is sometimes taken around the risky or stochastic steady state (e.g. [Coerudacier et al. \(2011\)](#), [Juillard \(2011\)](#)).

Suppose we want to take an approximation around an arbitrary point $[\check{s}, \check{p}, \check{b}, \check{f}]$. The strategy we suggest is to evaluate that point in each regime. More specifically we will have

$$\check{d}_{r_t} \equiv \tilde{d}_{r_t} \left(\check{b}, \check{f}, \check{s}, \check{p}, \check{b}, \check{f}, \check{p}, \check{b}, 0, \theta_{r_t} \right)$$

The quantity \check{d}_{r_t} , which is potentially different from zero is then forwarded to the first-order approximation when solving the first-order coefficients.

Interestingly, both the regime-specific approach and the ergodic approach are special cases. In the former, because $[\check{s}, \check{p}, \check{b}, \check{f}] = [s_t(r_t), p_t(r_t), b_t(r_t), f_t(r_t)]$, \check{d}_{r_t} is zero. In the later case, $[\check{s}, \check{p}, \check{b}, \check{f}] = [s^{ergodic}, p^{ergodic}, b^{ergodic}, f^{ergodic}]$ and \check{d}_{r_t} need not be zero.

The approach suggested here is computationally more efficient than that suggested by [Foerster et al. \(2014\)](#) and does not require any partitioning of the switching parameters between those that affect the steady state and those that do not. It will be shown later on in an application that we easily recover their results.

¹⁸In the RISE toolbox, this is triggered by the option “imposed”. When the steady state is not imposed, RISE uses the values provided as an initial guess in the solving of the approximation point.

3.2 First-order perturbation

At first order we seek to approximate \mathcal{T}^{r_t} in (3) with a solution of the form

$$\mathcal{T}^{r_t}(z) \simeq \mathcal{T}^{r_t}(\bar{z}_{r_t}) + \mathcal{T}_z^{r_t}(z_t - \bar{z}_{r_t}) \quad (8)$$

With the approximation point in hand, the only unknown at this stage is $\mathcal{T}_z^{r_t}$. Before finding its solution, we formally show that we can replace future parameters with auxiliary variables and thereby rationalize equation (5).

Proposition 1. *Let an equation in the system (2) be such that $aux_t - \theta_{r_t} = 0$, where aux_t is an endogenous variable and θ_{r_t} is a switching parameter. Suppose the model is approximated around an arbitrary point yielding the first-order expansion of aux_t as $aux_t = aux^* + \tau_x^{r_t} \hat{x}_{t-1} + \tau_\varepsilon^{r_t} \varepsilon_t + \tau_\sigma^{r_t} \sigma$. Then the following hold:*

- $\tau_x^{r_t} = 0$ and $\tau_\varepsilon^{r_t} = 0$
- $\tau_\sigma^{r_t} = \theta_{r_t} - aux^*$

Proof. The fact that $\tau_x^{r_t} = 0$ and $\tau_\varepsilon^{r_t} = 0$ is obvious because conditional on being in a regime r_t , $aux_t = \theta_{r_t}$ and is independent of any state variable. This leads to $aux_t = aux^* + \tau_\sigma^{r_t} \sigma$, implying that $aux^* + \tau_\sigma^{r_t} \sigma = \theta_{r_t}$ or $\tau_\sigma^{r_t} \sigma = \theta_{r_t} - aux^*$. Setting the perturbation parameter $\sigma = 1$ gives the result. \square

Finding the solution of $\mathcal{T}_z^{r_t}$ requires differentiating (7) with respect to z_t and keeping in mind that at the approximation point, \check{d}_{r_t} need not be equal to zero. Using tensor notation, we have

$$\left[\check{d}_{r_t} \right]^i + E_t \sum_{r_{t+1}=1}^h [d_v^{r_t, r_{t+1}}]_\alpha^i [v_z]_j^\alpha = 0 \quad (9)$$

where $[d_v^{r_t, r_{t+1}}]_\alpha^i$ denotes the derivative of the i^{th} row of d with respect to the α^{th} row of v and, similarly, $[v_z]_j^\alpha$ denotes the derivative of the α^{th} row of v with respect to the j^{th} row of z .

Unfolding the tensors this problem reduces to

$$\check{d}_{r_t} + \sum_{r_{t+1}=1}^h d_v^{r_t, r_{t+1}} E_t v_z = 0$$

Let us define $d_{g^q}^{r_t, r_{t+1}} \equiv \frac{\partial d^{r_t, r_{t+1}}}{\partial g^q}$ for $g = s, p, f, b$ referring to static, predetermined, forward looking and “both” variables and for $q = 0, +, -$ referring to current

variables, future variables and past variables respectively. Let us also define the coefficient matrix on contemporaneous variables as

$$A_{r_t, r_{t+1}}^0 \equiv \begin{bmatrix} d_{s^0}^{r_t, r_{t+1}} & d_{p^0}^{r_t, r_{t+1}} & d_{b^0}^{r_t, r_{t+1}} & d_{f^0}^{r_t, r_{t+1}} \end{bmatrix}$$

we have

$$dv = \begin{bmatrix} d_{b^+}^{r_t, r_{t+1}} & d_{f^+}^{r_t, r_{t+1}} & A_{r_t, r_{t+1}}^0 & d_{p^-}^{r_t, r_{t+1}} & d_{b^-}^{r_t, r_{t+1}} & d_{\varepsilon^0}^{r_t, r_{t+1}} & d_{\theta^+}^{r_t, r_{t+1}} \end{bmatrix}$$

The derivatives of v with respect to z are given by

$$v_z = a_z^0 + a_z^1 u \quad (10)$$

where the definitions of a_z^0 and a_z^1 are given in appendix (A.1).

An important ingredient in the calculation of v_z is, as can be seen in (6), the derivative of h with respect to z . This derivative is given by

$$h_z^{r_t} = \begin{bmatrix} (\lambda_x \mathcal{T}_z^{r_t})' & m'_\sigma & m'_{\varepsilon,1} & \cdots & m'_{\varepsilon,k} & 0_{n_z^2 \times n_\varepsilon} \end{bmatrix}'$$

With all these expressions in hand, the problem to solve for finding $\mathcal{T}_z^{r_t}$ can be expanded into

$$\sum_{r_{t+1}=1}^h \left(\begin{bmatrix} d_{b^+}^{r_t, r_{t+1}} & d_{f^+}^{r_t, r_{t+1}} \\ d_{p^-}^{r_t, r_{t+1}} & d_{b^-}^{r_t, r_{t+1}} \end{bmatrix} \begin{bmatrix} m_p \\ m_b \end{bmatrix} + d_{\varepsilon^0}^{r_t, r_{t+1}} m_{\varepsilon,0} + d_{\theta^+}^{r_t, r_{t+1}} \hat{\theta}_{r_{t+1}} m_\sigma \right) = 0$$

Looking at $\mathcal{T}_z^{r_t}$ and $h_z^{r_t}$ in detail, we see that they can be partitioned. In particular, with $\mathcal{T}_{z,x}^{r_t} \equiv \begin{bmatrix} \mathcal{T}_{z,p}^{r_t} & \mathcal{T}_{z,b}^{r_t} \end{bmatrix}$, we have

$$\mathcal{T}_z^{r_t} = \begin{bmatrix} \mathcal{T}_{z,x}^{r_t} & \mathcal{T}_{z,\sigma}^{r_t} & \mathcal{T}_{z,\varepsilon^0}^{r_t} & \mathcal{T}_{z,\varepsilon^1}^{r_t} & \cdots & \mathcal{T}_{z,\varepsilon^k}^{r_t} \end{bmatrix}$$

$$h_z^{r_t} = \begin{bmatrix} \lambda_x \mathcal{T}_{z,x}^{r_t} & \lambda_x \mathcal{T}_{z,\sigma}^{r_t} & \lambda_x \mathcal{T}_{z,\varepsilon^0}^{r_t} & \lambda_x \mathcal{T}_{z,\varepsilon^1}^{r_t} & \cdots & \lambda_x \mathcal{T}_{z,\varepsilon^k}^{r_t} \\ 0_{1 \times n_x} & 1 & 0_{1 \times n_\varepsilon} & 0_{1 \times n_\varepsilon} & \cdots & 0_{1 \times n_\varepsilon} \\ 0_{n_\varepsilon \times n_x} & 0_{n_\varepsilon \times 1} & 0_{n_\varepsilon} & I_{n_\varepsilon} & \cdots & 0_{n_\varepsilon} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{n_\varepsilon \times n_x} & 0_{n_\varepsilon \times 1} & 0_{n_\varepsilon} & 0_{n_\varepsilon} & \cdots & I_{n_\varepsilon} \\ 0_{n_\varepsilon \times n_x} & 0_{n_\varepsilon \times 1} & 0_{n_\varepsilon} & 0_{n_\varepsilon} & \cdots & 0_{n_\varepsilon} \end{bmatrix}$$

Hence the solving can be decomposed into small problems

3.2.1 Impact of endogenous state variables

The problem to solve is

$$A_{r_t}^0 \mathcal{T}_{z,x}^{r_t} + A_{r_t}^- + \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}} \lambda_x \mathcal{T}_{z,x}^{r_t} = 0 \quad (11)$$

with $A_{r_t}^0 \equiv \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^0$ and $A_{r_t}^- \equiv \sum_{r_{t+1}=1}^h \begin{bmatrix} d_{p^-}^{r_t, r_{t+1}} & d_{b^-}^{r_t, r_{t+1}} \end{bmatrix}$ and

$$A_{r_t, r_{t+1}}^+ \equiv \begin{bmatrix} 0_{n_d \times n_s} & 0_{n_d \times n_p} & d_{b^+}^{r_t, r_{t+1}} & d_{f^+}^{r_t, r_{t+1}} \end{bmatrix} \quad (12)$$

Since there are many algorithms for solving (11), we delay the presentation of our solution algorithms until Section 5. For the time being, the reader should note the way $A_{r_t, r_{t+1}}^+$ enters (11). This says that our algorithms will be able to handle cases where the coefficient matrix on forward-looking terms is known in the current period ($A_{r_t, r_{t+1}}^+ = A_{r_t, r_t}^+$) as in Farmer et al. (2011) but also the more complicated case where $A_{r_t, r_{t+1}}^+ \neq A_{r_t, r_t}^+$ as in Cho (2014). This is part of the reasons why the notation of Schmitt-Grohe and Uribe (2004), where one can stack variables, is not appropriate in this context. This assumption is very convenient in the Farmer et al. (2011) algorithm as it allows them to derive their solution algorithm, which would be more difficult otherwise. It is also convenient as it leads to substantial computational savings. But as our derivations show, the assumption is incorrect in problems where $A_{r_t, r_{t+1}}^+ \neq A_{r_t, r_t}^+$.

3.2.2 Impact of uncertainty

For the moment, we proceed with the assumption that we have solved for $\mathcal{T}_{z,x}^{r_t}$. Now we have to solve for $\mathcal{T}_{z,\sigma}^{r_t}$ using

$$\check{d}_{r_t} + \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,\sigma}^{r_{t+1}} + A_{\sigma}^{r_t} \mathcal{T}_{z,\sigma}^{r_t} + \sum_{r_{t+1}=1}^h d_{\theta^+}^{r_t, r_{t+1}} \hat{\theta}_{r_t, r_{t+1}} = 0$$

which leads to

$$\mathcal{T}_{z,\sigma}^{r_t} = - \begin{bmatrix} A_{\sigma}^1 + A_{1,1}^+ & A_{1,2}^+ & \cdots & A_{1,h}^+ \\ A_{2,1}^+ & A_{\sigma}^2 + A_{2,2}^+ & \cdots & A_{2,h}^+ \\ \vdots & \vdots & \ddots & \vdots \\ A_{h,1}^+ & A_{h,2}^+ & \cdots & A_{\sigma}^h + A_{h,h}^+ \end{bmatrix}^{-1} \begin{bmatrix} \check{d}_1 + \sum_{r_{t+1}=1}^h d_{\theta^+}^{1, r_{t+1}} \hat{\theta}_{1, r_{t+1}} \\ \check{d}_2 + \sum_{r_{t+1}=1}^h d_{\theta^+}^{2, r_{t+1}} \hat{\theta}_{2, r_{t+1}} \\ \vdots \\ \check{d}_h + \sum_{r_{t+1}=1}^h d_{\theta^+}^{h, r_{t+1}} \hat{\theta}_{h, r_{t+1}} \end{bmatrix} \quad (13)$$

where $A_{\sigma}^{r_t} \equiv A_{r_t}^0 + \sum_{r_{t+1}=1}^h \left(d_{f^+}^{r_t, r_{t+1}} \lambda_f \mathcal{T}_{z,p}^{r_{t+1}} \lambda_p + d_{b^+}^{r_t, r_{t+1}} \lambda_b \mathcal{T}_{z,b}^{r_{t+1}} \lambda_b \right)$.

It follows from equation (13) that in our framework, it is the presence of (1) future parameters in the current state system and/or (2) an approximation taken around a point that is not the regime-specific steady state that creates non-certainty equivalence.

3.2.3 Impact of shocks

Define

$$U_{r_t} \equiv \left(\sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}} \right) \lambda_x + A_{r_t}^0 \quad (14)$$

Contemporaneous shocks We have

$$\mathcal{T}_{z,\varepsilon^0}^{r_t} = -U_{r_t}^{-1} \sum_{r_{t+1}=1}^h d_{\varepsilon^0}^{r_t, r_{t+1}} \quad (15)$$

Future shocks For $k = 1, 2, \dots$ we have the recursive formula

$$\mathcal{T}_{z,\varepsilon^k}^{r_t} = -U_{r_t}^{-1} \times \left(\sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,\varepsilon^{(k-1)}}^{r_{t+1}} \right) \quad (16)$$

3.3 Second-order perturbation

The second-order perturbation solution of \mathcal{T}^{r_t} in (3) takes the form

$$\mathcal{T}^{r_t}(z) \simeq \mathcal{T}^{r_t}(\bar{z}_{r_t}) + \mathcal{T}_z^{r_t}(z_t - \bar{z}_{r_t}) + \frac{1}{2} \mathcal{T}_{zz}^{r_t}(z_t - \bar{z}_{r_t})^{\otimes 2}$$

Since $\mathcal{T}^{r_t}(\bar{z}_{r_t})$ and $\mathcal{T}_z^{r_t}$ have been computed in earlier steps, at this stage we only need to solve for $\mathcal{T}_{zz}^{r_t}$. To get the second-order solution, we differentiate (9) with respect to z to get

$$E_t \sum_{r_{t+1}=1}^h \left([d_{vv}^{r_t, r_{t+1}}]_{\alpha\beta}^i [v_z]_k^\beta [v_z]_j^\alpha + [d_v^{r_t, r_{t+1}}]_\alpha^i [v_{zz}]_{jk}^\alpha \right) = 0 \quad (17)$$

so that unfolding the tensors yields

$$\sum_{r_{t+1}=1}^h \left(d_{vv}^{r_t, r_{t+1}} E_t v_z^{\otimes 2} + d_v^{r_t, r_{t+1}} E_t v_{zz} \right) = 0 \quad (18)$$

We use the notation $A^{\otimes k}$ as a shorthand for $A \otimes A \otimes \dots \otimes A$. We get v_{zz} by differentiating v_z with respect to z , yielding

$$v_{zz} = a_{zz}^0 + a_{zz}^1 u^{\otimes 2} + a_{zz}^1 (u \otimes h_z^{r_t} + h_z^{r_t} \otimes u) \quad (19)$$

where the definitions of a_{zz}^0 , a_{zz}^1 as well as the expressions for $Ev_z^{\otimes 2}$, Ev_{zz} and $Eu^{\otimes 2}$ needed to solving for $\mathcal{T}_{zz}^{r_t}$ are given in appendix (A.2).

With those expressions in hand, expanding the problem to solve in (18) gives

$$A_{zz}^{r_t} + \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{zz}^{r_{t+1}} C_{zz}^{r_t} + U_{r_t} \mathcal{T}_{zz}^{r_t} = 0 \quad (20)$$

with $A_{zz}^{r_t} \equiv \sum_{r_{t+1}=1}^h d_{vv}^{r_t, r_{t+1}} E_t v_z^{\otimes 2}$ and $C_{zz}^{r_t} \equiv (h_z^{r_t})^{\otimes 2} + Eu^{\otimes 2}$.

3.4 Third-order perturbation

The third-order perturbation solution of \mathcal{T}^{r_t} in (3) takes the form

$$\mathcal{T}^{r_t}(z) \simeq \mathcal{T}^{r_t}(\bar{z}_{r_t}) + \mathcal{T}_z^{r_t}(z_t - \bar{z}_{r_t}) + \frac{1}{2} \mathcal{T}_{zz}^{r_t}(z_t - \bar{z}_{r_t})^{\otimes 2} + \frac{1}{3!} \mathcal{T}_{zzz}^{r_t}(z_t - \bar{z}_{r_t})^{\otimes 3}$$

Now the unknown is $\mathcal{T}_{zzz}^{r_t}$. We get its solution first by differentiating (17) with respect to z . This leads to

$$E_t \sum_{r_{t+1}=1}^h \left(\begin{array}{c} [d_{vv}^{r_t, r_{t+1}}]_{\alpha\beta\gamma}^i [v_z]_l^\gamma [v_z]_k^\beta [v_z]_j^\alpha + \\ [d_{vv}^{r_t, r_{t+1}}]_{\alpha\beta}^i \sum_{rst \in \Omega_1} [v_{zz}]_{rs}^\beta [v_z]_t^\alpha + \\ [d_v^{r_t, r_{t+1}}]_\alpha^i [v_{zzz}]_{jkl}^\alpha \end{array} \right) = 0$$

with $\Omega_1 \equiv \{klj, jlk, jkl\}$.

This tensor operation is unfolded into

$$E_t \sum_{r_{t+1}=1}^h (d_{vvv}^{r_t, r_{t+1}} (v_z^{\otimes 3}) + \omega_{\Omega_1} (d_{vv}^{r_t, r_{t+1}} (v_z \otimes v_{zz})) + d_v^{r_t, r_{t+1}} v_{zzz}) = 0$$

where $\omega_{\Omega_1}(\cdot)$ is a function that computes the sum of permutations of tensors of type $A(B \otimes C)$ and where the permutations are given by the indices in Ω_1 .

We get v_{zzz} by differentiating v_{zz} with respect to z , yielding

$$v_{zzz} = a_{zzz}^0 + a_{zzz}^1 P((h_z^{r_t})^{\otimes 2} \otimes u) + a_{zzz}^1 P(h_z^{r_t} \otimes u^{\otimes 2}) + a_{zzz}^1 u^{\otimes 3} + \omega_{\Omega_1}(a_{zz}^1 (u \otimes h_z^{r_t})) \quad (21)$$

where $h_{zz}^{r_t}$, defined in (33) is the second derivative of h^{r_t} with respect to z ¹⁹.

In equation (21), the function $P(\cdot)$ denotes the sum of all possible permutations of Kronecker products. That is $P(A \otimes B \otimes C) = A \otimes B \otimes C + A \otimes C \otimes B + B \otimes A \otimes C + B \otimes C \otimes A + C \otimes A \otimes B + C \otimes B \otimes A$. Only one product needs to be computed. The remaining ones can be derived as “perfect shuffles” of the computed one (see e.g. Van Loan (2000)).

The definitions of a_{zzz}^0 and a_{zzz}^1 as well as expressions for Ev_{zzz} , $E(v_z \otimes v_z \otimes v_z)$, $E(v_{zz} \otimes v_z)$, $E(v_z \otimes v_{zz})$ needed to solve for $\mathcal{T}_{zzz}^{r_t}$ are given in appendix (A.3). With all those expressions in hand, finding $\mathcal{T}_{zzz}^{r_t}$ reduces to solving

$$A_{zzz}^{r_t} + \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{zzz}^{r_{t+1}} C_{zzz}^{r_t} + U_{r_t} \mathcal{T}_{zzz}^{r_t} = 0 \quad (22)$$

with $A_{zzz}^{r_t} \equiv \sum_{r_{t+1}=1}^h \left\{ d_{vvv}^{r_t, r_{t+1}} E v_z^{\otimes 3} + \omega_{\Omega_1} \left(\begin{array}{c} d_{vv}^{r_t, r_{t+1}} E(v_z \otimes v_{zz}) + \\ \left(d_{b^+}^{r_t, r_{t+1}} \lambda_b + d_{f^+}^{r_t, r_{t+1}} \lambda_f \right) \mathcal{T}_{zz}^{r_{t+1}} (h_z^{r_t} \otimes h_{zz}^{r_t}) \end{array} \right) \right\}$
and $C_{zzz}^{r_t} \equiv P(h_z^{r_t} \otimes E u^{\otimes 2}) + E u^{\otimes 3} + (h_z^{r_t})^{\otimes 3}$.

3.5 Higher-order solutions

The derivatives of (7) can be generalized to an arbitrary order, say p . In that case we have

$$0 = E_t \sum_{r_{t+1}=1}^h \left\{ \sum_{l=1}^p [d_{v^{(l)}}^{r_t, r_{t+1}}]_{\alpha_1 \alpha_2 \dots \alpha_l}^i \sum_{c \in M_{l,p}} \prod_{m=1}^l [v_z^{|c_m|}]_{\mathbf{j}(c_m)}^{\alpha_m} \right\}$$

where:

- $M_{l,p}$ is the set of all partitions of the set of p indices with class l . In particular, $M_{1,p} = \{\{1, 2, \dots, p\}\}$ and $M_{p,p} = \{\{1\}, \{2\}, \dots, \{p\}\}$
- $|\cdot|$ denotes the cardinality of a set
- c_m is the m -th class of partition c
- $\mathbf{j}(c_m)$ is a sequence of j 's indexed by c_m

¹⁹For any $p \geq 2$, the p -order derivative of h^{r_t} with respect to z is given by $h_{z^{(p)}}^{r_t} = \begin{bmatrix} \lambda_x \mathcal{T}_{z^{(p)}}^{r_t} \\ 0_{(1+(k+1)n_\varepsilon) \times n_z^p} \end{bmatrix}$.

The p -order perturbation solution of \mathcal{T}^{r_t} takes the form

$$\begin{aligned} \mathcal{T}^{r_t}(z) \simeq & \mathcal{T}^{r_t}(\bar{z}_{r_t}) + \mathcal{T}_z^{r_t}(z_t - \bar{z}_{r_t}) + \frac{1}{2}\mathcal{T}_{zz}^{r_t}(z_t - \bar{z}_{r_t})^{\otimes 2} + \frac{1}{3!}\mathcal{T}_{zzz}^{r_t}(z_t - \bar{z}_{r_t})^{\otimes 3} + \\ & \frac{1}{4!}\mathcal{T}_{zzzz}^{r_t}(z_t - \bar{z}_{r_t})^{\otimes 4} + \dots + \frac{1}{p!}\mathcal{T}_{z\dots z}^{r_t}(z_t - \bar{z}_{r_t})^{\otimes p} \end{aligned} \quad (23)$$

4 Solving the generalized coupled Sylvester equation

The equations for finding the solution of higher-order terms in (20), in (22) and in all further orders of approximation can be, after preconditioning, written in a general form as $\tilde{X}_i + \sum_{j=1}^h F_{ij}\tilde{X}_j\tilde{G}_{ij} = \tilde{C}_i$ where \tilde{X}_j , $j = 1, 2, \dots, h$, are the unknown matrices of size $r_x \times c_x$.

In the case where $h = 1$, we have a simple Sylvester equation for which some efficient algorithms have already been derived, e.g. [Kamenik \(2005\)](#). Such algorithms cannot be applied in the switching case, unfortunately. However, there are instances in which the columns of \tilde{X}_j , are duplicated. RISE offers the option to exploit that symmetry by computing those terms only once. For that purpose, we need two matrices: a compressor matrix \mathcal{K} , of size $c_x \times c_k$, that will gather the unique columns of X_j , and an expander matrix \mathcal{E} , of size $c_k \times c_x$, that will undo the operation. Formally, $\tilde{X}_j = \tilde{X}_j\mathcal{K}\mathcal{E}$. The two matrices are such that $\mathcal{K}\mathcal{E}$ need not be equal to $I_{c_x \times c_x}$, but $\mathcal{E}\mathcal{K} = I_{c_k \times c_k}$. Hence, defining $X_j \equiv \tilde{X}_j\mathcal{K}$, we can solve for X_j and then expand the result into $\tilde{X}_j = X_j\mathcal{E}$.²⁰ In practice, rather than doing matrix multiplications, it is even more economical to implement \mathcal{K} as a selection vector and \mathcal{E} as a duplication vector. This is what RISE does. The compressed system is then

$$X_i + \sum_{j=1}^h F_{ij}X_jG_{ij} = C_i \quad (24)$$

where $G_{ij} \equiv \mathcal{E}\tilde{G}_{ij}\mathcal{K}$ and $C_i \equiv \tilde{C}_i\mathcal{K}$.

4.1 The direct approach

Define $X \equiv [X_1, X_2, \dots, X_h]$ and $C \equiv [C_1, C_2, \dots, C_h]$. The h equations can be stacked and vectorized yielding the direct solution

²⁰In a higher-order perturbation of order p , the number of unique columns of \tilde{X}_j is $c_k = \binom{c_x+p-1}{p}$.

$$x = \begin{bmatrix} I + G'_{11} \otimes F_{11} & \cdots & G'_{1h} \otimes F_{1h} \\ \vdots & \ddots & \vdots \\ G'_{h1} \otimes F_{h1} & \cdots & I + G'_{hh} \otimes F_{hh} \end{bmatrix}^{-1} c$$

where $x \equiv \text{vec}(X)$ and $c \equiv \text{vec}(C)$. The cost of building and inverting the matrix involved in the direct approach is huge and very often this approach is not feasible.

4.2 The iterative approach

The iterative approach avoids building the huge matrix in the direct approach.²¹ This can be done in different ways. By default, RISE applies the “transpose-free quasi minimum residuals” (TFQMR) method due to [Freund \(1993\)](#). This iterative procedure is defined by a quasi-minimization of the norm of a residual vector over a Krylov subspace. The advantage of this technique in addressing our problem is that it involves the huge coefficient matrix only in the form of matrix-vector products, which we can compute efficiently. Moreover, the TFQMR technique overcomes the problems of the bi-conjugate gradient (BCG) method, which is known to (i) exhibit erratic convergence behavior with large oscillations in the residual norm, (ii) have a substantial likelihood of breakdowns (more precisely, division by 0).²²

5 Solving the system of quadratic matrix equations

One of the main objectives of this paper is to present algorithms for solving (11), the first-order approximation of the RS-DSGE model. In this context, there are important contributions in the literature, most of which are ideally suited for small models. We briefly review the essence of some of the existing solution methods, most of which are nicely surveyed and evaluated in [Farmer et al. \(2011\)](#), and then present our own.

²¹In RISE the computation and storage of G_{ij} , which typically involves Kronecker products, is also avoided.

²²The BCG method, due to [Lanczos \(1952\)](#), extends the classical conjugate gradient algorithm (CG) to non-Hermitian matrices.

5.1 Existing solution methods

The FP algorithm: The strategy in this technique by [Farmer et al. \(2008\)](#) is to stack all the regimes and rewrite a MS-DSGE model as a constant parameter model, whose solution is the solution of the original model. As demonstrated by [Farmer et al. \(2011\)](#), the FP algorithm may not find solutions even when they exist.

The SW algorithm: This functional iteration technique by [Svensson and Williams \(2007\)](#) uses an iterative scheme to find the solution of a fix-point problem, through successive approximations of the solution, just as the [Farmer et al. \(2008\)](#) approach. But the SW algorithm does not resort to stacking matrices as [Farmer et al. \(2008\)](#), which is an advantage for solving large systems. Unfortunately, again as shown by [Farmer et al. \(2011\)](#), the SW algorithm may not find solutions even when they exist.

The FWZ algorithm: This algorithm, by [Farmer et al. \(2011\)](#), uses a Newton method to find the solutions of the MS-DSGE model. It is a robust procedure in the sense that it is able to find solutions that the other methods are not able to find. The Newton method has the advantage of being fast and locally stable around any given solution. However, unlike what was claimed in [Farmer et al. \(2011\)](#), in practice, choosing a large enough grid of initial conditions does not guarantee that one will find all possible MSV solutions.

The Cho algorithm: This algorithm, by [Cho \(2014\)](#), finds the so-called “forward solution” by extending the forward method of [Cho and Moreno \(2011\)](#) to MS-DSGE models. The paper also addresses the issue of determinacy – existence of a unique stable solution. The essence of the algorithm is the same as that of the SW algorithm.

The FRWZ algorithm²³: This technique, by [Foerster et al. \(2013, 2014\)](#), is one of the latest in the literature. It applies the theory of Gröbner bases (see [Buchberger \(1965, 2006\)](#)) to find “all possible” solutions and then establishes determinacy by checking whether there is a unique Mean-Square Stable (MSS) solution²⁴. As argued earlier, because of their complexity, the Gröbner bases avenue is not very attractive in the context of models of the size routinely used in central banks. Moreover,

²³This algorithm is not implemented in RISE, but the flexibility and modularity of the toolbox are such that any user with little exposure to RISE is able to apply his own solution method without having to change anything to the core codes. Therefore the FRWZ algorithm can easily be incorporated as long as it is written in Matlab or interfaced to Matlab.

²⁴A formal definition of this stability concept is given in Section 5.3.

the determinacy-checking step is as computationally intensive as the first step, if not more so. The matrix in equation (30) with its Kronecker products has to be formed explicitly and then its eigenvalues have to be computed. In fact, for all the applications we consider in this paper, checking MSS is the most expensive step. If one attempts estimating a model, which is the goal we have in mind when deriving the algorithms in this paper, using the Foerster et al. (2013, 2014) strategy is prohibitively expensive : locating the global peak of the posterior kernel will typically require thousands of function evaluations and for each of those function evaluations we need to solve a problem of exponential complexity.

Finally, stability of a first-order approximation does not imply stability of higher-order approximations even in the constant-parameter case. In the switching case with constant transition probabilities, the problem gets worse. When we allow for endogenous probabilities as it is done in this paper, there is no stability concept such as MSS that applies. Therefore, provided we can even compute all possible solutions, we do not have a criterion for picking one of them.

5.2 Proposed solution techniques

All of the algorithms presented above use constant transition probabilities. In many cases, they also use the type of notation used in Klein (2000), Schmitt-Grohe and Uribe (2004) and Sims (2002). This notation works well with constant-parameter models.²⁵ When it comes to RS-DSGE models, however, that notation has some disadvantages as we argued earlier. Among other things, besides the fact that it often requires considerable effort to manipulate the model into the required form, the ensuing algorithms may be slow, owing in part to the creation of additional auxiliary variables.

The solution methods proposed below do not suffer from those drawbacks, but have their own weaknesses as well. And so they should be viewed as complementary to existing ones.²⁶

5.2.1 A functional iteration algorithm

Functional iteration is a well established technique for solving models that do not have an analytical solution, but that are nevertheless amenable to a fix-point type of representation. It is very much used for instance when solving value functions in the

²⁵This is an attractive feature as it provides a compact and efficient representation of the first-order conditions or constraints of a system leading to the well known solution methods for constant-parameter DSGE models such as Klein (2000), Sims (2002), among others.

²⁶These solution algorithms have been part of the RISE toolbox for many years.

Bellman equations of dynamic programming problems. In the context of RS-DSGE models, functional iteration has the advantages that (1) it typically converges fast whenever it is able to find a solution, (2) can be used to solve large systems, (3) is easily implementable.

We concentrate on minimum state variable (MSV) solutions and promptly rewrite (11) as²⁷

$$\mathcal{T}_{z,x}^{r_t} = - \left[A_{r_t}^0 + \left(\sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}} \right) \lambda_x \right]^{-1} A_{r_t}^- \quad (25)$$

which is the basis for our functional iteration algorithm whose pseudo code is given in algorithm 1.

<p>Data: $\mathcal{T}_{z,x}^{(0)} \equiv \left[\mathcal{T}_{z,x}^{1(0)} \quad \dots \quad \mathcal{T}_{z,x}^{h(0)} \right]$, $A_{r_t, r_{t+1}}^+$, $A_{r_t}^0$, $A_{r_t}^-$, for $r_t, r_{t+1} = 1, 2, \dots, h$</p> <p>Result: $\mathcal{T}_{z,x}^{(k)}$</p> <pre> 1 k=0 ; 2 set conv=inf ; 3 while conv > 0 do 4 k=k+1 ; 5 for r_t = 1 to h do 6 tmp = A_{r_t}^0 ; 7 for r_{t+1} = 1 to h do 8 tmp = tmp + A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}(k-1)} ; 9 end 10 \mathcal{T}_{z,x}^{r_t(k)} = - [tmp]^{-1} A_{r_t}^- ; 11 end 12 conv = \left\ \mathcal{T}_{z,x}^{(k)} - \mathcal{T}_{z,x}^{(k-1)} \right\ ; 13 end </pre>
--

Algorithm 1: Pseudo code for the generic Functional iteration algorithm

This algorithm is implemented in RISE under the name **MFI_full**. It has a variant that further exploits the sparsity brought about by the existence of static variables. This version of the algorithm is called **MFI**.

²⁷The invertibility of $A_{r_t}^0 + \left(\sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}} \right) \lambda_x$ is potentially reinforced by the fact that all variables in the system appear as current. That is, all columns of matrix $A_{r_t}^0$ have at least one non-zero element.

Local convergence of functional iteration Now we are interested in deriving the conditions under which an initial guess will converge to a solution. We have the following theorem:

Theorem 2. *Let $r_t = 1, 2, \dots, h$ and $r_{t+1} = 1, 2, \dots, h$ be indices over the number of regimes. Let $T_{r_t}^*$, be a solution to (11), such that $\|\lambda_x T_{r_t}^*\| \leq \vartheta$ and $\|\tilde{A}_{r_t, r_{t+1}}^+\| \leq a^+$ and $\left\| \left(A_{r_t}^0 + \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ T_{r_{t+1}}^* \lambda_x \right)^{-1} \right\| \leq \tau$ where $A_{r_t, r_{t+1}}^+ \equiv p_{r_t, r_{t+1}} \tilde{A}_{r_t, r_{t+1}}^+$, $\vartheta > 0$, $a^+ > 0$, $\tau > 0$ and with $p_{r_t, r_{t+1}}$ the (steady-state) transition probability of going from regime r_t to regime r_{t+1} . Assume there exists $\delta > 0$ such that $\|T_{r_t}^{(0)} - T_{r_t}^*\| \leq \delta$, $r_t = 1, 2, \dots, h$. Then if $\tau a^+ (\vartheta + \delta) < 1$, the iterative sequence $\left\{ T_{r_t}^{(k)} \right\}_{r_t=1}^h$ generated by functional iteration with initial guess $T_{r_t}^{(0)}$, $r_t = 1, 2, \dots, h$, satisfies $\|T_{r_t}^{(k)} - T_{r_t}^*\| \leq \varphi \sum_{r_{t+1}=1}^h p_{r_t, r_{t+1}} \|T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^*\|$, with $\varphi \in (0, 1)$.*

Proof. See appendix B. □

Farmer et al. (2011) show that functional iterations are not able to find all possible solutions. But still, functional iterations cannot be discounted: they are useful for getting solutions for large systems. It is therefore important to understand why in some cases functional iteration may fail to converge.

The theorem establishes some sufficient conditions for convergence of functional iteration in this context. In other words it gives us hints about the cases in which functional iteration might not perform well. In particular, inequality $\tau a^+ (\vartheta + \delta) < 1$ is more likely to be fulfilled for low values of τ , a^+ , ϑ and δ rather than large ones.

5.2.2 Newton algorithms

We define

$$W_{r_t}(\mathcal{T}_{z,x}) \equiv \mathcal{T}_{z,x}^{r_t} + \left[A_{r_t}^0 + \left(\sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}} \right) \lambda_x \right]^{-1} A_{r_t}^- \quad (26)$$

If the current guess $\mathcal{T}_{z,x}^{r_t}$ is not such that $W_{r_t}(\mathcal{T}_{z,x}) = 0$ for $r_t = 1, 2, \dots, h$, we need to improve the guess. We do this by perturbing the current guess $\mathcal{T}_{z,x}$ by a factor Δ and then expanding (26) into

$$W_{r_t}(\mathcal{T}_{z,x} + \Delta) = W_{r_t}(\mathcal{T}_{z,x}) + \Delta_{r_t} - \left(\sum_{r_{t+1}=1}^h L_{x^+}^{r_t, r_{t+1}} \Delta_{r_{t+1}} \right) L_{x^-}^{r_t} + HOT$$

where²⁸

$$L_{x^+}^{r_t, r_{t+1}} \equiv U_{r_t}^{-1} A_{r_t, r_{t+1}}^+ \quad (27)$$

and

$$L_{x^-}^{r_t} \equiv -\lambda_x \mathcal{T}_{z,x}^{r_t} \quad (28)$$

Newton ignores *HOT* and attempts to solve for Δ_{r_t} so that

$$W_{r_t}(\mathcal{T}_{z,x}) + \Delta_{r_t} - \left(\sum_{r_{t+1}=1}^h L_{x^+}^{r_t, r_{t+1}} \Delta_{r_{t+1}} \right) L_{x^-}^{r_t} = 0 \quad (29)$$

where $\Delta_{r_t} - \left(\sum_{r_{t+1}=1}^h L_{x^+}^{r_t, r_{t+1}} \Delta_{r_{t+1}} \right) L_{x^-}^{r_t}$ is the Frechet derivative of W_{r_t} at $\mathcal{T}_{z,x}$ in direction Δ_{r_t} .

Equation (29) is similar to (24) if we let $i = r_t$, $j = r_{t+1}$, $C_i = -W_{r_t}(\mathcal{T}_{z,x})$, $F_{ij} = L_{x^+}^{r_t, r_{t+1}}$ and $G_{ij} = -L_{x^-}^{r_t}$ and so can be solved for Δ_{r_t} using the same techniques. This gives rise to the two main Newton algorithms of RISE: **MNK_full** and **MN_full**. The former uses a direct approach for solving the Newton step, while the latter uses the iterative approach. Their variants that exploit the sparsity stemming from the presence of static variables are called **MNK** and **MN** respectively.

Relation to Farmer, Waggoner and Zha (2011) As discussed earlier, [Farmer et al. \(2011\)](#) consider problems in which the coefficient matrix on the forward-looking variables is known in the current period. Our algorithms consider the more general case where those matrices are unknown in the current period. Another difference between [Farmer et al. \(2011\)](#) and this paper is that rather than perturbing the $\mathcal{T}_{z,x}$ as we do, they build the Newton by computing derivatives directly. Our approach is more efficient as it solves a smaller system. In particular, because their algorithm is a direct extension of [Sims \(2002\)](#), [Farmer et al. \(2011\)](#) also have to solve for expectational errors in addition to the other variables in the system. By building the Newton directly through derivatives, they fail to recognize that the problem of finding the Newton step can recast into solving a system of generalized coupled Sylvester equations. As a result they have to use the direct approach in section (4.1), which builds a huge matrix going through Kronecker products and then inverts the matrix to find the solution.

²⁸*HOT* are the higher-order terms of the approximation.

Our Newton solution strategies can be summarized in the pseudo code given in algorithm 2.

<pre> Data: $\mathcal{T}_{z,x}^{(0)} \equiv \left[\mathcal{T}_{z,x}^{1(0)} \quad \dots \quad \mathcal{T}_{z,x}^{h(0)} \right]$, $A_{r_t, r_{t+1}}^+$, $A_{r_t}^0$, $A_{r_t}^-$, for $r_t, r_{t+1} = 1, 2, \dots, h$ Result: $\mathcal{T}_{z,x}^{(k)}$ 1 k=0 ; 2 compute $W^{(k)}(\mathcal{T}_{z,x}) \equiv \left[W_1(\mathcal{T}_{z,x}^{(k)}) \quad \dots \quad W_h(\mathcal{T}_{z,x}^{(k)}) \right]$, using (26) ; 3 while $W^{(k)} \neq 0$ do 4 for $r_t = 1$ to h do 5 for $r_{t+1} = 1$ to h do 6 compute $L_{x^+}^{r_t, r_{t+1}}$ as in (27); 7 end 8 compute $L_{x^-}^{r_t}$ as in (28); 9 end 10 solve (29) for $\Delta^{(k)} \equiv \left[\Delta_1^{(k)} \quad \dots \quad \Delta_h^{(k)} \right]$ using the results in section (4.1) or those in section (4.2); 11 k=k+1 ; 12 $\mathcal{T}_{z,x}^{(k)} = \mathcal{T}_{z,x}^{(k-1)} + \Delta^{(k-1)}$; 13 for $r_t = 1$ to h do 14 $tmp = A_{r_t}^0$; 15 for $r_{t+1} = 1$ to h do 16 $tmp = tmp + A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}(k-1)}$; 17 end 18 $W_{r_t}(\mathcal{T}_{z,x}^{(k)}) = \mathcal{T}_{z,x}^{r_t(k)} + [tmp]^{-1} A_{r_t}^-$; 19 end 20 end </pre>
--

Algorithm 2: Pseudo code for the generic Newton algorithm

5.3 Stability of first-order approximation under constant transition probabilities

The traditional stability concepts for constant-parameter linear rational expectations models do not extend to the regime switching case. Following the lead of Svensson and Williams (2007) and Farmer et al. (2011) among others, this paper uses the concept of mean square stability (MSS), to characterize stable solutions of non-time-

varying transition probabilities.

Consider the RS-DSGE system whose solution is given by equation (8) and with constant transition probability matrix Q such that $Q_{r_t, r_{t+1}} = p_{r_t, r_{t+1}}$. Using the selection matrix λ_x , we can extract the portion of the solution for state endogenous variables only to get

$$x_t(z) = \lambda_x \mathcal{T}^{r_t}(\bar{z}_{r_t}) + \lambda_x \mathcal{T}_{z,x}^{r_t}(x_{t-1} - \lambda_x \mathcal{T}^{r_t}(\bar{z}_{r_t})) + \lambda_x \mathcal{T}_{z,\sigma}^{r_t} \sigma + \lambda_x \mathcal{T}_{z,\varepsilon^0}^{r_t} \varepsilon_t + \dots + \lambda_x \mathcal{T}_{z,\varepsilon^k}^{r_t} \varepsilon_{t+k}$$

This system and thereby (8) is MSS if for any initial condition x_0 , there exist a vector μ and a matrix Σ independent of x_0 such that $\lim_{t \rightarrow \infty} \|Ex_t - \mu\| = 0$ and $\lim_{t \rightarrow \infty} \|Ex_t x_t' - \Sigma\| = 0$. Hence the covariance matrix of the process is bounded. As shown by [Gupta et al. \(2003\)](#) and ([Costa et al., 2005](#), chapter 3), a necessary and sufficient condition for MSS is that matrix Υ , as defined in (30), has all its eigenvalues inside the unit circle²⁹.

$$\Upsilon \equiv (Q \otimes I_{n_x^2 \times n_x^2}) \begin{bmatrix} \lambda_x \mathcal{T}_{z,x}^1 \otimes \lambda_x \mathcal{T}_{z,x}^1 & & & \\ & \ddots & & \\ & & \lambda_x \mathcal{T}_{z,x}^h \otimes \lambda_x \mathcal{T}_{z,x}^h & \end{bmatrix} \quad (30)$$

A particular note should be taken of the facts that:

- With endogenous probabilities as this paper discusses, the MSS concept becomes useless.
- Even in the constant-transition probabilities case, there are no theorems implying that stability of the first-order perturbation implies the stability of higher orders.

5.4 Proposed initialization strategies

Both functional iteration and the Newton methods require an initial guess of the solution. As stated earlier, we do not attempt to compute all possible solutions of (11) and then check that only one of them is stable, as is done in [Foerster et al.](#)

²⁹It is not very hard to see that a computationally more efficient representation of Υ given by:

$$\begin{bmatrix} p_{1,1} (\lambda_x \mathcal{T}_{z,x}^1 \otimes \lambda_x \mathcal{T}_{z,x}^1) & p_{1,2} (\lambda_x \mathcal{T}_{z,x}^2 \otimes \lambda_x \mathcal{T}_{z,x}^2) & \cdots & p_{1,h} (\lambda_x \mathcal{T}_{z,x}^h \otimes \lambda_x \mathcal{T}_{z,x}^h) \\ p_{2,1} (\lambda_x \mathcal{T}_{z,x}^1 \otimes \lambda_x \mathcal{T}_{z,x}^1) & p_{2,2} (\lambda_x \mathcal{T}_{z,x}^2 \otimes \lambda_x \mathcal{T}_{z,x}^2) & \cdots & p_{2,h} (\lambda_x \mathcal{T}_{z,x}^h \otimes \lambda_x \mathcal{T}_{z,x}^h) \\ \vdots & \vdots & & \vdots \\ p_{h,1} (\lambda_x \mathcal{T}_{z,x}^1 \otimes \lambda_x \mathcal{T}_{z,x}^1) & p_{h,2} (\lambda_x \mathcal{T}_{z,x}^2 \otimes \lambda_x \mathcal{T}_{z,x}^2) & \cdots & p_{h,h} (\lambda_x \mathcal{T}_{z,x}^h \otimes \lambda_x \mathcal{T}_{z,x}^h) \end{bmatrix}.$$

(2013, 2014). Initialization is key not only to finding a solution but also, potentially, for the speed of convergence of any algorithm.

We derive all of our initialization schemes from making assumptions on equation (25), which, for convenience we report in (31), abstracting from the iteration superscript.

$$\mathcal{T}_{z,x}^{r_t} = \left[A_{r_t}^0 + \left(\sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}} \right) \lambda_x \right]^{-1} A_{r_t}^- \quad (31)$$

there are at least three ways to initialize a solution:

- The first approach makes the assumption that $A_{r_t}^- = 0$, $r_t = 1, 2, \dots, h$. In this case then, the initial guess is the solution that we would obtain in the absence of predetermined variables. This solution is trivially zero as can be seen from equation (31). Formally, it corresponds to

$$\mathcal{T}_{z,x}^{r_t(0)} = 0, \text{ for } r_t = 1, 2, \dots, h$$

- The second approach makes the assumption that $A_{r_t, r_{t+1}}^+ = 0$, $r_t, r_{t+1} = 1, 2, \dots, h$. While the first approach assumes there are no predetermined variables, this approach instead assumes there are no forward-looking variables. In that case, the solution is also easy to compute using (31). In RISE, this option is the default initialization scheme. It corresponds to setting:

$$\mathcal{T}_{z,x}^{r_t(0)} = - [A_{r_t}^0]^{-1} A_{r_t}^-, \text{ for } r_t = 1, 2, \dots, h$$

- The third approach recognizes that making a good guess for a solution is difficult and instead tries to make guesses on the products $A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}}$. We proceed as follows: (i) we would like the guesses for each regime to be related, just as they would be in the final solution; (ii) we would like the typical element of $A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}}$ to have a norm roughly of the squared order of magnitude of a solvent or final solution. Given those two requirements, we make the assumption that the elements of $A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}}$, $r_t, r_{t+1} = 1, 2, \dots, h$ are drawn from a standard normal distribution scaled by the approximate norm of a solvent squared.³⁰ We approximate the norm of a solvent by

$$\sigma \equiv \frac{\eta_0 + \sqrt{\eta_0^2 + 4\eta_+ \eta_-}}{2\eta_+}$$

³⁰Implicit also is the assumption that $A_{r_t, r_{t+1}}^+$ and $\mathcal{T}_{z,x}^{r_{t+1}}$ are of the same order of magnitude.

where $\eta_0 \equiv \max_{r_t} (\|A_{r_t}^0\|)$, $\eta_- \equiv \max_{r_t} (\|A_{r_t}^-\|)$, $\eta_+ \equiv \max_{r_t} (\|A_{r_t, r_{t+1}}^+\|)$. Then with draws of $A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}}$ in hand, we can use (31) to compute a guess for a solution. In RISE this approach is used when trying to locate alternative solutions. It can formally be expressed as

$$T_{r_t}^{(0)} = - \left(A_{r_t}^0 + \left(\sum_{r_{t+1}=1}^h [A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}}]^{guess} \right) \lambda_x \right)^{-1} A_{r_t}^-$$

with $[A_{r_t, r_{t+1}}^+ \mathcal{T}_{z,x}^{r_{t+1}}]_{ij}^{guess} \sim \sigma^2 N(0, 1)$.

6 Performance of alternative solution methods

In this section, we investigate the performance of different algorithms. These include our functional iteration algorithms **MFI_full** and **MFI**; our Newton algorithms **MNK_full**, **MNK**, **MN_full** and **MN**; and the Newton algorithm by [Farmer et al. \(2011\)](#), which we call **FWZ**.³¹ The FWZ algorithm is included for direct comparison as it was shown to outperform the FP and SW algorithms described above. We assess the performance of those algorithms in terms of their ability to find solutions but also in terms of their efficiency. To that end, we use a set of problems found in the literature.

6.1 Test problems

The details of the test problems considered are given in appendix (C). The models considered are the following:

1. **cho214_EIS** is a six-variable model by [Cho \(2014\)](#), with switching elasticity of intertemporal substitution.
2. **cho214_MP** is a six-variable model by [Cho \(2014\)](#), with switching monetary policy parameters.
3. **frwz2013_nk_1** is a three-variable New Keynesian model by [Foerster et al. \(2013\)](#)
4. **frwz2013_nk_2** is the same model as **frwz2013_nk_1** but with a different parameterization.

³¹This algorithm is also included among the solvers in RISE and the coding strictly follows the formulae in the [Farmer et al. \(2011\)](#) paper.

5. **frwz2013_nk_hab** is a four-variable New Keynesian model with habit persistence by [Foerster et al. \(2013\)](#).
6. **frwz2014_rbc** is a three-variable RBC model by [Foerster et al. \(2014\)](#)
7. **fwz2011_1** is a two-variable model by [Farmer et al. \(2011\)](#)
8. **fwz2011_2** is the same model as **fwz2011_1** but with a different parameterization.
9. **fwz2011_3** is the same model as **fwz2011_1** but with a different parameterization.
10. **smets_wouters** is the well-known model by [Smets and Wouters \(2007\)](#). Originally the model is a constant-parameter model. We turn it into a switching model by creating a regime in which monetary policy is unresponsive to changes in inflation and output gap. See Appendix (C) for further details.

All the tests use the same tolerance level for all the algorithms and are run on a *Dell Precision M4700 computer with Intel Core i7 processor running at 2.8 GHz with 16GB of RAM*. Windows 7 (64-bit) is the operating system used. The results included in this paper are computed using MATLAB 8.4.0.150421 (R2014b) environment.

6.2 Finding a solution: computational efficiency

The purpose of this experiment is to assess both the ability of each algorithm to find a solution and the speed with which the solution is found. To that end, we solved each model 50 times for each solution procedure and averaged the computing time spent across runs.³² The computation times we report include:

- the time spent in finding a solution,
- the time spent in checking whether the solution is MSS, which is oftentimes the most costly step.³³
- some further overhead from various functions that call the solution routines in the object oriented system.

³²Running the same code several times is done simply to smooth out the uncertainty across runs.

³³As argued earlier, the matrix in equation (30) with its Kronecker products has to be formed explicitly and then its eigenvalues have to be computed.

The last two elements are constant across models and therefore should not have an impact on the relative performances of the algorithms.

The results presented in Table 1 are all computed using the default initialization scheme of RISE, i.e. the one that assumes there are no forward-looking variables in the model.³⁴ The table reports for each model its relative speed performance. This relative performance is computed as the absolute performance divided by the best performance. A relative performance of say, x , means that for that particular model, the algorithm is x times slower than the best, whose performance is always 1. The table also reports in column 2 the number of variables in the model under consideration and in column 3 the speed in seconds of the fastest solver.

The speed of computation tends to increase with the size of the model as could have been expected, although this pattern is not uniform for small models. But we note that the MNK_full algorithm tends to be the fastest algorithm on small models. The FWZ algorithm comes first in two of the contests but as argued earlier because it imposes that the coefficient matrix on forward-looking variables depends only on the current regime, it saves some computations and gives the wrong answer when the coefficient matrix on forward-looking variables also depends on the future regime. This is the case in the Cho2014_EIS model.

The MNK_full algorithm is also faster than its version that exploits the sparsity of the problem, namely MNK. This is not surprising however because when the number of variables is small, the overhead incurred in trying to exploit sparsity outweighs the benefits. This is also true for the functional iteration algorithms as MFI_full tends to dominate MFI.

For the same small models, the MNK_full algorithm, which uses Kronecker products, is also faster than its counterparts that do not use Kronecker products, namely MN_full and MN. Again, here there is no surprise: when the number of variables is small, computing Kronecker products and inverting small matrices is faster than trying to avoid those computations, which is the essence of the MN_full and MN algorithms.

The story does not end there, however. As the number of variables increases, the MN algorithm shows better performance. By the time the number of variables gets to 40 as in the Smets and Wouters model, the MN algorithm, which exploits sparsity and avoid the computation of Kronecker products, becomes the fastest. In particular, on the smets.wouters model, the MN algorithm is about 7 times faster than its Kronecker version (MNK), which also exploits sparsity and about 39 times faster than the version of the algorithm that builds Kronecker products and does not exploit

³⁴The results for other initialization schemes have also been computed and are available upon request.

sparsity (MNK_full), which is the fastest on small models. For the parameterization under consideration, the FWZ and the functional iteration algorithms never find a solution.³⁵

Table 1: Relative efficiency of various algorithms

	nvars	best speed	mfi	mnk	mn	mfi_full	mnk_full	mn_full	fwz
cho2014_EIS	6	0.01798	4.971	1.624	2.802	3.365	1.177	2.126	1
cho2014_MP	6	0.01259	1	1.221	1.101	1.11	1.124	1.082	Inf
frwz2013_nk_1	3	0.01486	1.34	1.198	1.459	1.219	1	1.435	1.092
frwz2013_nk_2	3	0.01571	1.309	1.111	1.42	1.218	1	1.407	1.059
frwz2013_nk_hab	4	0.01622	1.059	1.106	1.134	1	1.009	1.125	1.146
frwz2014_rbc	3	0.01628	2.676	1.259	1.313	2.15	1	1.284	Inf
fwz2011_1	2	0.00924	Inf	1.051	1.139	Inf	1.015	1.068	1
fwz2011_2	2	0.01043	1.57	1.281	1.374	1.27	1	1.265	1.278
fwz2011_3	2	0.01108	1.877	1.185	1.35	1.443	1	1.279	1.112
smets_wouters	40	0.2515	Inf	6.514	1	Inf	39.48	1.364	Inf

The table reports for each model (rows), the relative speed performance of each algorithm (columns 4 to 10) computed as the absolute performance divided by the best performance. A relative performance of say, x , means that for that particular model, the algorithm is x times slower than the best, whose performance is always 1. The table also reports in column 2 the number of variables in the model under consideration and in column 3 the speed in seconds of the fastest solver. The computations are done using the “backward” initialization scheme of RISE.

6.3 Finding many solutions

Our algorithms are not designed to find all possible solutions since the results they produce depend on the initial condition. Although sampling randomly may lead to different solutions, many different starting points may also lead to the same solution. Despite these shortcomings, we investigated the ability of the algorithms to find many solutions for the test problems considered. Unlike Foerster et al. (2013, 2014) our

³⁵Although the Newton algorithms converge fast, rapid convergence occurs only in the neighborhood of a solution (see for instance Benner and Byers (1998)). It may well be the case that the first Newton step is disastrously large, requiring many iterations to find the region of rapid convergence. This is an issue we have not encountered so far, but it may be addressed by incorporating line searches along a search direction so as to reduce excessively long steps or increase excessively short ones, which in both cases would accelerate convergence even further.

algorithms only produce real solutions, which are the ones of interest. Starting from 100 different random points sampled using the third initialization scheme described in Section 5.4, we computed the results given in Table 2. For every combination of solution algorithm and test problem, there are two numbers. The first one is the total number of solutions found and the second one is the number of stable solutions. As the results show, we easily replicate the findings in the literature. In particular,

- for the [Cho \(2014\)](#) problems, we replicate the finding that the model with switching elasticity of intertemporal substitution is determinate. Indeed all of the algorithms tested find at least one solution and at most one of the solutions found is stable in the MSS sense. We do not, however, find that the solution found for the model with switching monetary policy is stable.
- for the [Farmer et al. \(2011\)](#) examples, we replicate all the results. In the first parameterization, there is a unique solution and the solution is stable. This is what all the Newton algorithms find. The functional iteration algorithms are not able to find that solution. In the second parameterization, our Newton algorithms return two stable solutions and in the third parameterization, they return four stable solutions just as found by [Farmer et al. \(2011\)](#).
- [Foerster et al. \(2013\)](#) examples: In the New Keynesian model without habits our Newton algorithms return 3 real solutions out of which only one is stable. In the second parameterization, they return three real solutions but this time two of them are stable. For the New Keynesian model with habits, our Newton algorithms return six real solutions out of which one is stable.
- for the [Foerster et al. \(2014\)](#) RBC model, there is a unique stable solution.
- the [Smets and Wouters \(2007\)](#) model, shows that there are many real solutions and many of those solutions are stable. The different algorithms return different solutions since they are all initialized randomly in different places. This suggests that we most likely have not found all possible solutions and that by increasing the number of simulations we could have found many more solutions.

6.4 Higher orders

Our algorithms do not just solve higher order approximations for RS-DSGE models. They solve higher-order approximations of constant-parameter DSGE models as a special case. We have tested our algorithms on various models against some of

Table 2: Solutions found after 100 random starting values

	nvars	mfi	mnk	mn	mfi_full	mnk_full	mn_full	fwz
cho2014_EIS	6	1 1	2 1	2 1	1 1	2 1	2 1	2 1
cho2014_MP	6	1 0	1 0	1 0	1 0	1 0	1 0	0 0
frwz2013_nk_1	3	1 1	3 1	3 1	1 1	3 1	3 1	3 1
frwz2013_nk_2	3	1 1	3 2	3 2	1 1	3 2	3 2	3 2
frwz2013_nk_hab	4	1 1	6 1	6 1	1 1	6 1	6 1	6 1
frwz2014_rbc	3	1 1	1 1	1 1	1 1	2 1	2 1	0 0
fwz2011_1	2	0 0	1 1	1 1	0 0	1 1	1 1	1 1
fwz2011_2	2	1 1	2 2	2 2	1 1	2 2	2 2	1 1
fwz2011_3	2	1 1	4 4	4 4	1 1	4 4	4 4	3 3
smets_wouters	40	0 0	11 2	11 4	0 0	16 2	10 3	0 0

The table reports for each model (rows), the performance of each solution algorithm (columns 3 to 9). Each entry is a vector of two numbers. The first element in the vector is the number of real solutions found and the second element the number of stable (MSS) real solutions.

the existing and freely available algorithms solving for higher-order approximations. Those algorithms include : dynare, dynare++ and codes by [Binning \(2013a,b\)](#). In all cases we find the same results.

We have successfully solved a second-order approximation of a version of NEMO, the DSGE model used by Norges Bank. This model includes upwards of 272 equations in its nonlinear form.

We have run the Markov-switching RBC model by [Foerster et al. \(2014\)](#) in RISE and the results for a third-order approximation are presented below. Because in [Foerster et al. \(2014\)](#) the steady state is unique, we simply had to pass this information to the software, without any further need to partition the parameters. RISE prints only the unique combination of cross products and uses the sign @sig to denote the perturbation term. Only the state variables with non-zero coefficients across the endogenous variables are reported. The first block reports the solution of the model in the first regime and the second block, the solution in the second regime. In each block, the second, third and fourth column represent the solution in levels for consumption (C), capital (K) and the technology process (Z) respectively. The steady state of those variables is given right below each variable name, in the line labeled “steady state”. Below that line are the coefficients on state variables expressed as deviations from their steady states as in equation (23). The results produced by RISE closely replicate the second-order approximation presented by [Foerster et al.](#)

(2014).³⁶

MODEL SOLUTION
SOLVER :: mfi

Regime 1 : a = 1 & const = 1

Endo Name	C	K	Z
steady state	2.082588	22.150375	1.007058
K{-1}	0.040564	0.969201	0.000000
Z{-1}	0.126481	-2.140611	0.100000
@sig	-0.009066	-0.362957	0.018459
EPS	0.009171	-0.155212	0.007251
K{-1},K{-1}	-0.000461	-0.000167	0.000000
K{-1},Z{-1}	0.001098	-0.047836	0.000000
K{-1},@sig	-0.000462	-0.008170	0.000000
K{-1},EPS	0.000080	-0.003468	0.000000
Z{-1},Z{-1}	-0.058668	1.168197	-0.044685
Z{-1},@sig	0.000323	0.018293	0.000916
Z{-1},EPS	0.000299	0.007642	0.000360
@sig,@sig	-0.024022	0.026995	0.000169
@sig,EPS	0.000023	0.001326	0.000066
EPS,EPS	0.000022	0.000554	0.000026
K{-1},K{-1},K{-1}	0.000011	0.000005	0.000000
K{-1},K{-1},Z{-1}	-0.000009	0.000001	0.000000
K{-1},K{-1},@sig	-0.000001	-0.000000	0.000000
K{-1},K{-1},EPS	-0.000001	0.000000	0.000000
K{-1},Z{-1},Z{-1}	-0.000332	0.017407	0.000000
K{-1},Z{-1},@sig	0.000005	0.000269	0.000000
K{-1},Z{-1},EPS	0.000002	0.000115	0.000000
K{-1},@sig,@sig	-0.000185	0.000231	0.000000
K{-1},@sig,EPS	0.000000	0.000020	0.000000
K{-1},EPS,EPS	0.000000	0.000008	0.000000
Z{-1},Z{-1},Z{-1}	0.037330	-0.811474	0.028102
Z{-1},Z{-1},@sig	-0.000250	-0.006515	-0.000273
Z{-1},Z{-1},EPS	-0.000097	-0.002777	-0.000107
Z{-1},@sig,@sig	-0.000418	-0.000480	0.000006
Z{-1},@sig,EPS	-0.000001	-0.000043	0.000002
Z{-1},EPS,EPS	0.000001	-0.000018	0.000001
@sig,@sig,@sig	-0.002111	0.001640	0.000001
@sig,@sig,EPS	-0.000028	-0.000038	0.000000
@sig,EPS,EPS	-0.000000	-0.000003	0.000000
EPS,EPS,EPS	0.000000	-0.000001	0.000000

Regime 2 : a = 2 & const = 1

Endo Name	C	K	Z
steady state	2.082588	22.150375	1.007058
K{-1}	0.040564	0.969201	0.000000
@sig	-0.100434	0.926307	-0.041021
EPS	0.026867	-0.464994	0.021752
K{-1},K{-1}	-0.000461	-0.000167	0.000000
K{-1},@sig	-0.001160	0.020327	0.000000
K{-1},EPS	0.000233	-0.010399	0.000000
@sig,@sig	-0.023223	0.043449	0.000835
@sig,EPS	-0.000525	-0.009756	-0.000443

³⁶The second order terms of the RISE output should be multiplied by 2 in order to compare them with the output presented in Foerster et al. (2014).

EPS, EPS	0.000187	0.004982	0.000235
K_{-1}, K_{-1}, K_{-1}	0.000011	0.000005	0.000000
K_{-1}, K_{-1}, σ	0.000007	-0.000003	0.000000
$K_{-1}, K_{-1}, \text{EPS}$	-0.000002	0.000000	0.000000
K_{-1}, σ, σ	-0.000188	0.000476	0.000000
$K_{-1}, \sigma, \text{EPS}$	-0.000002	-0.000146	0.000000
$K_{-1}, \text{EPS}, \text{EPS}$	0.000001	0.000075	0.000000
σ, σ, σ	-0.001411	0.002643	-0.000011
$\sigma, \sigma, \text{EPS}$	-0.000077	-0.000228	0.000006
$\sigma, \text{EPS}, \text{EPS}$	-0.000002	0.000069	-0.000003
EPS, EPS, EPS	0.000001	-0.000036	0.000002

Finally we can use the same [Foerster et al. \(2014\)](#) model to illustrate in a third-order approximation, the effect of technology shock that is announced four periods from now. As argued earlier and as shown in our derivations, this does not require changing the model file: the model is solved directly under the assumption that agents have information about future shocks. We compare two economies. In the first economy, agents fully anticipate the shock and therefore internalize its effects already in the first period. In the second economy, agents will be aware of the shock only when it will materialize four periods from now. As [figure 1](#) illustrates, consumption already increases today while capital falls in the “anticipate” scenario.

7 Conclusion

In this paper we have proposed new solution algorithms for solving regime-switching DSGE models. The algorithms developed are useful for analyzing economic data subject to breaks and shifts in a rational expectations environment, where agents, while forming their expectations, explicitly take into account the probability of a regime shift. In the process we have derived higher-order perturbations of RS-DSGE models in which the probability of switching can be endogenous and in which anticipated future events can affect the current behavior of economic agents.

Despite the many benefits of departing from the assumption of constant parameters, the techniques and the ideas of RS-DSGE modeling have yet to be fully adopted by many macroeconomists as is the case for constant-parameter DSGE models popularized through Dynare. Two main reasons could potentially explain this. On the one hand, such models are a lot more difficult to solve and more computationally intensive than their constant-parameter counterparts and on the other hand, there are not many flexible and easy-to-use tools that would allow economists unable or unwilling to write complex code to take full advantage of these techniques and address current policy questions. By providing both efficient solution algorithms and

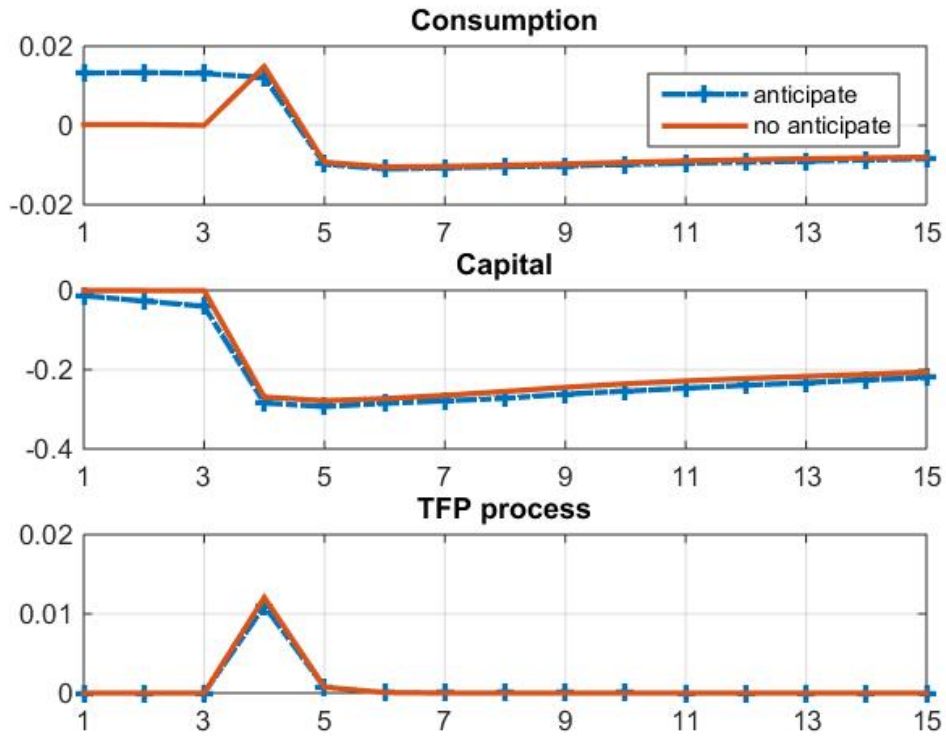


Figure 1: Third-order perturbation impulse responses to a TFP shock in the FRWZ2014_rbc model

the implementation of those algorithms in a toolbox, we hope to have assuaged these problems.

References

- Adjemian, S., Bastani, H., Juillard, M., Karame, F., Mihoubi, F., Perendia, G., Pfeifer, J., Ratto, M., and Villemot, S. (2011). Dynare: Reference manual, version 4. Dynare Working Papers 1, CEPREMAP.
- Aruoba, S. B., Cuba-Borda, P., and Schorfheide, F. (2014). Macroeconomic dynamics near the ZLB: A tale of two countries. PIER Working Paper 14-035, University of Pennsylvania.

- Barthelemy, J. and Marx, M. (2011). State-dependent probability distributions in non linear rational expectations models. Working Paper 347, Banque de France.
- Benner, P. and Byers, R. (1998). An exact line search method for solving generalized continuous-time algebraic riccati equations. *IEEE Transactions on Automatic Control*, 43(1):101–107.
- Bernanke, B. S., Gertler, M., and Gilchrist, S. (1999). The financial accelerator in a quantitative business cycle framework. In Taylor, J. B. and Woodford, M., editors, *Handbook of Macroeconomics*, volume 1, chapter 21, pages 1341–1393. Elsevier, 1 edition.
- Bi, H. and Traum, N. (2013). Estimating fiscal limits: The case of Greece. *Journal of Applied Econometrics(forthcoming)*.
- Binning, A. J. (2013a). Solving second and third-order approximations to DSGE models: A recursive sylvester equation solution. Working Paper 2013—18, Norges Bank.
- Binning, A. J. (2013b). Third-order approximation of dynamic models without the use of tensors. Working Paper 2013—13, Norges Bank.
- Buchberger, B. (1965). *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck.
- Buchberger, B. (2006). *An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal*. PhD thesis, RISC (Research Institute for Symbolic Computation), Johannes Kepler University Linz.
- Campbell, J. R., Fisher, J. D., and Justiniano, A. (2012). Monetary policy forward guidance and the business cycle. Mimeo, Federal Reserve Bank of Chicago.
- Canny, J. and Emiris, I. Z. (1993). An efficient algorithm for the sparse mixed resultant. *Lecture Notes in Computer Science*, 673:89–104.
- Cho, S. (2014). Characterizing Markov-Switching Rational Expectation Models. Working paper, Yonsei University.
- Cho, S. and Moreno, A. (2011). The forward method as a solution refinement in rational expectations models. *Journal of Economic Dynamics and Control*, 35(3):257–272.

- Coerudacier, N., Rey, H., and Winant, P. (2011). The risky steady-state. *American Economic Review*, 101(3):398–401.
- Cogley, T., Sargent, T. J., and Surico, P. (2012). The return of the Gibson paradox. Working paper, New York University.
- Costa, O. L. D. V., Fragoso, M. D., and Marques, R. P. (2005). *Discrete-Time Markov Jump Linear Systems*. Springer.
- Davig, T. and Leeper, E. M. (2007). Fluctuating macro policies and the fiscal theory. In *NBER Macroeconomics Annual 2007*, volume 21, pages 247–316. National Bureau of Economic Research.
- Davig, T., Leeper, E. M., and Walker, T. B. (2011). Inflation and the fiscal limit. *European Economic Review*, 55(1):31–47.
- Del Negro, M., Giannoni, M., and Patterson, C. (2013). The forward guidance puzzle. Staff Reports 574, Federal Reserve Bank of New York.
- Dreesen, P., Batselier, K., and De Moor, B. (2012). Back to the roots: Polynomial system solving, linear algebra, systems theory. *Proc 16th IFAC Symposium on System Identification (SYSID 2012)*, pages 1203–1208.
- Emiris, I. Z. (1996). On the complexity of sparse elimination. *Journal of Complexity*, 12:134–166.
- Emiris, I. Z. and Mourrain, B. (1999). Matrices in elimination theory. *Journal of Symbolic Computation*, 28(1-2):3–44.
- Farmer, R. F., Waggoner, D. F., and Zha, T. (2008). Minimal State Variable Solutions to Markov Switching Rational Expectations Models. Working Paper 2008-23, Federal Reserve Bank of Atlanta.
- Farmer, R. F., Waggoner, D. F., and Zha, T. (2011). Minimal State Variable Solutions to Markov Switching Rational Expectations Models. *Journal of Economics Dynamics and Control*, 35(12):2150–2166.
- Foerster, A., Rubio-Ramirez, J., Waggoner, D., and Zha, T. (2013). Perturbation Methods for Markov-Switching Models. Working Paper 2013-1, Federal Reserve Bank of Atlanta.

- Foerster, A., Rubio-Ramirez, J., Waggoner, D., and Zha, T. (2014). Perturbation Methods for Markov-Switching Models. Working Paper 20390, National Bureau of Economic Research.
- Freund, R. W. (1993). A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems. *SIAM Journal of Scientific Computing*, 2(14):470–482.
- Garcia, C. B. and Zangwill, W. I. (1979). Finding all solutions to polynomial systems and other systems of equations. *Mathematical Programming*, 16:159–176.
- Gavin, W. T., Keen, B. D., Richter, A. W., and Throckmorton, N. A. (2014). The limitations of forward guidance. Working Paper 2013-038C, Federal Reserve Bank of St. Louis.
- Gupta, V., Murray, R., and Hassibi, B. (2003). On the control of jump linear markov systems with markov state estimation. In *Proceedings of the 2003 American Automatic Control Conference*, pages 2893–2898.
- Juillard, M. (2011). Local Approximation of DSGE Models around the Risky Steady State. Mimeo, Banque de France.
- Juillard, M. and Maih, J. (2010). Estimating DSGE models with observed real-time expectation data. Mimeo, Norges Bank.
- Justiniano, A. and Primiceri, G. E. (2008). The time-varying volatility of macroeconomic fluctuations. *American Economic Review*, 98(3):604–41.
- Kamenik, O. (2005). Solving sdge models: A new algorithm for the Sylvester equation. *Computational Economics*, 25(1):167–187.
- Klein, P. (2000). Using the generalized schur form to solve a multivariate linear rational expectations model. *Journal of Economic Dynamics and Control*, 24:1405–1423.
- Lanczos, C. (1952). Solution of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, 49:33–53.
- Levintal, O. (2014). Fifth order perturbation solution to DSGE models. Working paper, Interdisciplinary Center Herzliya (IDC), School of Economics.
- Lubik, T. A. and Schorfheide, F. (2004). Testing for indeterminacy: An application to u.s. monetary policy. *American Economic Review*, 94(1):190–217.

- Maih, J. (2010). Conditional forecasts in DSGE models. Working Paper 2010—07, Norges Bank.
- Morgan, A. and Sommese, A. (1987). Computing all solutions to polynomial systems using homotopy continuation. *Applied Mathematics and Computation*, 24(2):115–138.
- Richter, A. W., Throckmorton, N. A., and Walker, T. B. (2014). Accuracy speed and robustness of policy function iteration. *Computational Economics*, 44(4):445–476.
- Ritt, J. F. (2008). Differential algebra. In Society, A. M., editor, *Colloquium Publications*, volume 33.
- Schmitt-Grohe, S. and Uribe, M. (2004). Solving dynamic general equilibrium models using a second-order approximation to the policy function. *Journal of Economics Dynamics and Control*, 28:755–775.
- Sims, C. A. (2002). Solving linear rational expectations models. *Computational Economics*, 20(1):1–20.
- Sims, C. A. and Zha, T. (2006). Were there regime switches in US monetary policy? *American Economic Review*, 96(1):54–81.
- Smets, F. and Wouters, R. (2007). Shocks and frictions in US business cycles: A bayesian dsge approach. *American Economic Review, American Economic Association*, 97(3):586–606.
- Stock, J. H. and Watson, M. W. (2003). Has the business cycle changed? evidence and explanations. *FRB Kansas City symposium, Jackson Hole*.
- Svensson, L. E. O. and Williams, N. (2007). Monetary policy with model uncertainty: Distribution forecast targeting. CEPR Discussion Papers 6331, C.E.P.R.
- Svensson, L. E. O. and Williams, N. (2009). Optimal monetary policy under uncertainty in dsge models: A markov jump-linear-quadratic approach. In Schmidt-Hebbel, K., Walsh, C. E., and Loayza, N., editors, *Central Banking, Analysis, and Economic Policies*, volume 13 of *Monetary Policy under Uncertainty and Learning*, pages 077–114. Central Bank of Chile.
- Van Loan, C. F. (2000). The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123(123):85–100.

Wu, W. T. (1978). On the decision problem and the mechanization of theorem-proving in elementary geometry. *Scientia Sinica*, 21:159–172.

A Expressions for various expectations terms

A.1 First order

The definitions of a_z^0 and a_z^1 appearing in equation (10) are as follows

$$a_z^0 \equiv \begin{pmatrix} \lambda_{bf} \mathcal{T}_z^{r_{t+1}} h_z^{r_t} \\ \mathcal{T}_z^{r_t} \\ m_p \\ m_b \\ m_{\varepsilon,0} \\ \hat{\theta}_{r_{t+1}} m_\sigma \end{pmatrix} \quad a_z^1 \equiv \begin{pmatrix} \lambda_{bf} \mathcal{T}_z^{r_{t+1}} \\ 0_{(n_s+2n_p+2n_b+n_f+n_\varepsilon+n_\theta) \times n_z} \end{pmatrix}$$

We have

$$Ev_z = a_z^0 \quad (32)$$

A.2 Second order

$$h_{zz}^{r_t} = \begin{bmatrix} \lambda_x \mathcal{T}_{zz}^{r_t} \\ 0_{(1+(k+1)n_\varepsilon) \times n_z^2} \end{bmatrix} \quad (33)$$

The definitions of a_{zz}^0 and a_{zz}^1 appearing in equation (19) are as follows:

$$a_{zz}^0 \equiv \begin{pmatrix} \lambda_{bf} \mathcal{T}_{z,x}^{r_{t+1}} \lambda_x \mathcal{T}_{zz}^{r_t} + \lambda_{bf} \mathcal{T}_{zz}^{r_{t+1}} (h_z^{r_t})^{\otimes 2} \\ \mathcal{T}_{zz}^{r_t} \\ 0_{(n_x+n_\varepsilon+n_\theta) \times n_z^2} \end{pmatrix} \quad a_{zz}^1 \equiv \begin{pmatrix} \lambda_{bf} \mathcal{T}_{zz}^{r_{t+1}} \\ 0 \\ 0 \end{pmatrix}$$

It follows that

$$Ev_{zz} = a_{zz}^0 + a_{zz}^1 Eu^{\otimes 2} \quad (34)$$

and

$$Ev_z^{\otimes 2} = (a_z^0)^{\otimes 2} + (a_z^1)^{\otimes 2} Eu^{\otimes 2} \quad (35)$$

The definition of u in (4) implies that³⁷

³⁷In general, we will have $E(u^{\otimes p}) = \begin{bmatrix} 0_{(n_p+n_b+1+kn_\varepsilon)^p \times n_z^p} \\ \text{vec}(M_p) m_\sigma^{\otimes p} \end{bmatrix}$, where M_p is the tensor of the p-order moments of ε_t . This expression should be cheap to compute since m_σ is a sparse vector.

$$E(u^{\otimes 2}) = \begin{bmatrix} 0_{(n_p+n_b+1+kn_\varepsilon)^2 \times n_z^2} \\ \tilde{I}_{n_\varepsilon}(m_\sigma \otimes m_\sigma) \end{bmatrix}$$

A.3 Third order

The definitions of a_{zzz}^0 and a_{zzz}^1 appearing in equation (21) are as follows:

$$a_{zzz}^0 \equiv \begin{pmatrix} \lambda_{bf} \mathcal{T}_{zzz}^{r_{t+1}}(h_z^{r_t})^{\otimes 3} + \omega_1 (\lambda_{bf} \mathcal{T}_{zz}^{r_{t+1}}(h_z^{r_t} \otimes h_{zz}^{r_t})) + \lambda_{bf} \mathcal{T}_z^{r_{t+1}} h_{zzz}^{r_t} \\ \mathcal{T}_{zzz}^{r_t} \\ 0_{(n_x+n_\varepsilon+n_\theta) \times n_z^3} \end{pmatrix} \quad (36)$$

$$a_{zzz}^1 \equiv \begin{pmatrix} \lambda_{bf} \mathcal{T}_{zzz}^{r_{t+1}} \\ 0_{(n_T+n_x+n_\varepsilon+n_\theta) \times n_z^3} \end{pmatrix} \quad (37)$$

where

$$h_{zzz}^{r_t} = \begin{bmatrix} \lambda_x \mathcal{T}_{zzz}^{r_t} \\ 0_{(1+(k+1)n_\varepsilon) \times n_z^3} \end{bmatrix}$$

$$\frac{\partial^3 \mathcal{T}^{r_{t+1}}(h^r(z)+uz)}{\partial z^3} = \mathcal{T}_{zzz}^{r_{t+1}}(h_z^{r_t})^{\otimes 3} + \omega_1 (\mathcal{T}_{zz}^{r_{t+1}}(h_z^{r_t} \otimes h_{zz}^{r_t})) + \mathcal{T}_{z,x}^{r_{t+1}} \lambda_x \mathcal{T}_{zzz}^{r_t} + \mathcal{T}_{zzz}^{r_{t+1}} u^{\otimes 3} + \omega_1 (\mathcal{T}_{zz}^{r_{t+1}}(u \otimes h_{zz}^{r_t})) + \mathcal{T}_{zzz}^{r_{t+1}} [P((h_z^{r_t})^{\otimes 2} \otimes u) + P(h_z^{r_t} \otimes u^{\otimes 2})]$$

We have

$$Ev_{zzz} = a_{zzz}^0 + a_{zzz}^1 P(h_z^{r_t} \otimes Eu^{\otimes 2}) + a_{zzz}^1 Eu^{\otimes 3} \quad (38)$$

We also need $E(v_z^{\otimes 3})$, $E(v_z \otimes v_{zz})$. Using (19) and (10), those quantities are computed as

$$Ev_z^{\otimes 3} = \begin{aligned} & (a_z^0)^{\otimes 3} + Ea_z^0 \otimes (a_z^1 u)^{\otimes 2} + \\ & E(a_z^1 u)^{\otimes 2} \otimes a_z^0 + E(a_z^1 u) \otimes a_z^0 \otimes (a_z^1 u) + E(a_z^1 u)^{\otimes 3} \end{aligned} \quad (39)$$

$$E(v_z \otimes v_{zz}) = \begin{aligned} & a_z^0 \otimes a_{zz}^0 + (a_z^1 \otimes a_{zz}^1) (Eu^{\otimes 2} \otimes h_z^{r_t} + E(u \otimes h_z^{r_t} \otimes u)) + \\ & a_z^0 \otimes a_{zz}^1 Eu^{\otimes 2} + (a_z^1 \otimes a_{zz}^1) Eu^{\otimes 3} \end{aligned} \quad (40)$$

B Local convergence of functional iteration

Before proving the theorem we first present a Banach perturbation lemma that will facilitate the subsequent derivations.

Lemma 3. *Banach Perturbation Lemma:* Let Ξ be a non-singular matrix. If Ω is such that $\|\Xi^{-1}\| \|\Omega\| < 1$, then $\|(\Xi + \Omega)^{-1}\| \leq \frac{\|\Xi^{-1}\|}{1 - \|\Xi^{-1}\| \|\Omega\|}$

Proof. We have

$$\begin{aligned} (\Xi + \Omega)^{-1} &= [\Xi (I + \Xi^{-1}\Omega)]^{-1} \\ &= \left[\sum_{i=0}^{\infty} (-\Xi^{-1}\Omega)^i \right] \Xi^{-1} \end{aligned}$$

which implies that

$$\begin{aligned} \|(\Xi + \Omega)^{-1}\| &\leq \left\| \sum_{i=0}^{\infty} (-\Xi^{-1}\Omega)^i \right\| \|\Xi^{-1}\| \\ &\leq \sum_{i=0}^{\infty} \|(-\Xi^{-1}\Omega)^i\| \|\Xi^{-1}\| \\ &= \sum_{i=0}^{\infty} \|(\Xi^{-1}\Omega)^i\| \|\Xi^{-1}\| \\ &= \frac{1}{1 - \|(\Xi^{-1}\Omega)\|} \|\Xi^{-1}\| \\ &\leq \frac{\|\Xi^{-1}\|}{1 - \|\Xi^{-1}\| \|\Omega\|} \end{aligned}$$

□

Now we can prove the theorem. Define

$$\Xi_{r_t} \equiv A_{r_t}^0 + \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ T_{r_{t+1}}^* \lambda_x$$

and

$$\Omega_{r_t}^{(k-1)} \equiv \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ (T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^*) \lambda_x$$

which implies

$$\begin{aligned} \left\| \Omega_{r_t}^{(k-1)} \right\| &= \left\| \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ (T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^*) \lambda_x \right\| \\ &\leq a^+ \sum_{r_{t+1}=1}^h p_{r_t, r_{t+1}} \left\| T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^* \right\| \end{aligned} \quad (41)$$

Also, we can write

$$T_{r_t}^{(k)} - T_{r_t}^{(*)} = \left(\Omega_{r_t}^{(k-1)} + \Xi_{r_t} \right)^{-1} \sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \left[T_{r_{t+1}}^* - T_{r_{t+1}}^{(k-1)} \right] \lambda_x T_{r_t}^{(*)}$$

so that

$$\begin{aligned}
\left\| T_{r_t}^{(k)} - T_{r_t}^{(*)} \right\| &\leq \left\| \left(\Omega_{r_t}^{(k-1)} + \Xi_{r_t} \right)^{-1} \right\| \left\| \left[\sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \left(T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^{(*)} \right) \right] \right\| \left\| \lambda_x T_{r_t}^{(*)} \right\| \\
&= \left\| \lambda_x T_{r_t}^{(*)} \right\| \left\| \left(\Omega_{r_t}^{(k-1)} + \Xi_{r_t} \right)^{-1} \right\| \left\| \left[\sum_{r_{t+1}=1}^h A_{r_t, r_{t+1}}^+ \left(T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^{(*)} \right) \right] \right\| \\
&\leq \left\| \lambda_x T_{r_t}^{(*)} \right\| \left\| \left(\Omega_{r_t}^{(k-1)} + \Xi_{r_t} \right)^{-1} \right\| a^+ \sum_{r_{t+1}=1}^h p_{r_t, r_{t+1}} \left\| \left(T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^{(*)} \right) \right\| \\
&\leq \vartheta a^+ \left\| \left(\Omega_{r_t}^{(k-1)} + \Xi_{r_t} \right)^{-1} \right\| \sum_{r_{t+1}=1}^h p_{r_t, r_{t+1}} \left\| \left(T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^{(*)} \right) \right\|
\end{aligned} \tag{42}$$

We can now use induction in the rest of the proof. Suppose $\left\| T_{r_t}^{(0)} - T_{r_t}^* \right\| \leq \delta$, $r_t = 1, 2, \dots, h$. Then (41) implies $\left\| \Omega_{r_t}^{(0)} \right\| \leq a^+ \delta$. Hence $\left\| \Xi_{r_t}^{-1} \right\| \left\| \Omega_{r_t}^{(0)} \right\| \leq \tau a^+ \delta$. Note that the requirement that $\tau a^+ (\vartheta + \delta) < 1$ automatically implies that $\tau a^+ \delta < 1$. Therefore we can use the perturbation lemma to establish that $\left\| \left(\Omega_{r_t}^{(0)} + \Xi_{r_t} \right)^{-1} \right\| \leq \frac{\left\| \Xi_{r_t}^{-1} \right\|}{1 - \left\| \Xi_{r_t}^{-1} \right\| \left\| \Omega_{r_t}^{(0)} \right\|} \leq \frac{\tau}{1 - \tau a^+ \delta}$. Plugging this result into (42) implies that $\left\| T_{r_t}^{(1)} - T_{r_t}^{(*)} \right\| \leq \frac{\vartheta a^+ \tau}{1 - \tau a^+ \delta} \sum_{r_{t+1}=1}^h p_{r_t, r_{t+1}} \left\| \left(T_{r_{t+1}}^{(0)} - T_{r_{t+1}}^{(*)} \right) \right\|$, essentially defining $\varphi \equiv \frac{\vartheta a^+ \tau}{1 - \tau a^+ \delta}$ and saying that the proposition holds for $k = 1$.

Now suppose that the proposition holds for some arbitrary order k , that is $\left\| T_{r_t}^{(k)} - T_{r_t}^* \right\| \leq \varphi \sum_{r_{t+1}=1}^h p_{r_t, r_{t+1}} \left\| T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^* \right\|$. We now have to show that it also holds for order $k+1$. Since by assumption $\left\| \left(T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^* \right) \right\| \leq \delta$, for $r_t = 1, 2, \dots, h$, we have $\left\| T_{r_t}^{(k)} - T_{r_t}^* \right\| \leq \varphi \sum_{r_{t+1}=1}^h p_{r_t, r_{t+1}} \left\| T_{r_{t+1}}^{(k-1)} - T_{r_{t+1}}^* \right\| \leq \varphi \delta \leq \delta$. (41) implies $\left\| \Omega_{r_t}^{(k)} \right\| \leq a^+ \delta$. Then by the perturbation lemma, it follows that $\left\| \left(\Omega_{r_t}^{(k)} + \Xi_{r_t} \right)^{-1} \right\| \leq \frac{\tau}{1 - \tau a^+ \delta}$. This implies by (42) that

$$\begin{aligned}
\left\| T_{r_t}^{(k+1)} - T_{r_t}^{(*)} \right\| &\leq \vartheta a^+ \left\| \left(\Omega_{r_t}^{(k)} + \Xi_{r_t} \right)^{-1} \right\| \sum_{r_{t+1}=1}^h p_{r_t, r_{t+1}} \left\| \left(T_{r_{t+1}}^{(k)} - T_{r_{t+1}}^{(*)} \right) \right\| \\
&\leq \frac{\vartheta a^+ \tau}{1 - \tau a^+ \delta} \sum_{r_{t+1}=1}^h p_{r_t, r_{t+1}} \left\| \left(T_{r_{t+1}}^{(k)} - T_{r_{t+1}}^{(*)} \right) \right\|
\end{aligned}$$

And so, the proposition also holds at order $k+1$.

C Models

C.1 Cho 2014: Markov switching elasticity of intertemporal substitution

C.1.1 Model equations

$$\begin{aligned}
\pi_t - \kappa y_t &= \beta \pi_{t+1} + z_{S,t} \\
y_t + \frac{1}{\sigma(s_t)} i_t &= \frac{1}{\sigma(s_t)} \pi_t + \frac{\sigma(s_{t+1})}{\sigma(s_t)} y_t + \frac{1}{\sigma(s_t)} z_{D,t} \\
-(1 - \rho) \phi_\pi \pi_t + i_t &= \rho i_{t-1} + z_{MP,t} \\
z_{S,t} &= \rho_s z_{S,t-1} + \epsilon_t^S, & \epsilon_t^S &\sim N(0, 1) \\
z_{D,t} &= \rho_d z_{D,t-1} + \epsilon_t^D, & \epsilon_t^D &\sim N(0, 1) \\
z_{MP,t} &= \rho_{mp} z_{MP,t-1} + \epsilon_t^{MP}, & \epsilon_t^{MP} &\sim N(0, 1)
\end{aligned}$$

C.1.2 Parameterization

ρ	ϕ_π	$\sigma(1)$	$\sigma(2)$	κ	β	ρ_s	ρ_d	ρ_{mp}	$p_{1,1}$	$p_{2,2}$
0.95	1.5	1	5	0.132	0.99	0.95	0.95	0	0.95	0.875

C.2 Cho 2014: Regime switching monetary policy

C.2.1 Model equations

$$\begin{aligned}
\pi_t - \kappa y_t &= \beta \pi_{t+1} + z_{S,t} \\
y_t + \frac{1}{\sigma} i_t &= \frac{1}{\sigma} \pi_t + y_t + z_{D,t} \\
-(1 - \rho) \phi_\pi \pi_t + i_t &= \rho i_{t-1} + z_{MP,t} \\
z_{S,t} &= \rho_s z_{S,t-1} + \epsilon_t^S, & \epsilon_t^S &\sim N(0, 1) \\
z_{D,t} &= \rho_d z_{D,t-1} + \epsilon_t^D, & \epsilon_t^D &\sim N(0, 1) \\
z_{MP,t} &= \rho_{mp} z_{MP,t-1} + \epsilon_t^{MP}, & \epsilon_t^{MP} &\sim N(0, 1)
\end{aligned}$$

C.2.2 Parameterization

ρ	$\phi_\pi(1)$	$\phi_\pi(2)$	σ	κ	β	ρ_s	ρ_d	ρ_{mp}	$p_{1,1}$	$p_{2,2}$
0.95	0.9	1.5	1	0.132	0.99	0	0	0	0.85	0.95

C.3 Farmer, Waggoner and Zha (2011)

C.3.1 Model equations

$$\begin{aligned} \phi\pi_t &= \pi_{t+1} + \delta\pi_{t-1} + \beta r_t \\ r &= \rho r_{t-1} + \varepsilon_t \end{aligned} \quad , \quad \varepsilon_t \sim N(0, 1)$$

C.3.2 Parameterization

	$\delta(1)$	$\delta(2)$	$\beta(1)$	$\beta(2)$	$\rho(1)$	$\rho(2)$	$\phi(1)$	$\phi(2)$	$p_{1,1}$	$p_{2,2}$
parameterization 1	0	0	1	1	0.9	0.9	0.5	0.8	0.8	0.9
parameterization 2	-0.7	0.4	1	1	0	0	0.5	0.8	0	0.64
parameterization 3	-0.7	-0.2	1	1	0	0	0.2	0.4	0.9	0.8

C.4 Foerster, Rubio-Ramirez, Waggoner and Zha (2014): RBC model

C.4.1 Model equations

$$\begin{aligned} c_t^{v-1} &= \beta z_t^{v-1} c_{t+1}^{v-1} (\alpha z_{t+1}^{1-\alpha} k_t^{\alpha-1} + 1 - \delta) \\ c_t + z_t k_t &= z_t^{1-\alpha} k_{t-1}^\alpha + (1 - \delta) k_{t-1} \\ \log(z_t) &= (1 - \rho(s_t)) \mu(s_t) + \rho(s_t) \log(z_{t-1}) + \sigma(s_t) \varepsilon_t \quad , \quad \varepsilon_t \sim N(0, 1) \end{aligned}$$

C.4.2 Steady state

$$\begin{aligned} z_t &= \exp(\mu) \\ k_t &= \left[\frac{1}{\alpha z_t^{1-\alpha}} \left(\frac{1}{\beta z_t^{v-1}} - 1 + \delta \right) \right]^{\frac{1}{\alpha-1}} \\ c_t &= z_t^{1-\alpha} k_t^\alpha + (1 - \delta) k_t - z_t k_t \end{aligned}$$

C.4.3 Parameterization

α	β	v	δ	$\mu(1)$	$\mu(2)$	$\rho(1)$	$\rho(2)$	$\sigma(1)$	$\sigma(2)$	$p_{1,1}$	$p_{2,2}$
0.33	0.9976	-1	0.025	0.0274	-0.0337	0.1	0	0.0072	0.0216	0.75	0.5

C.5 Foerster, Rubio-Ramirez, Waggoner and Zha (2013): NK model

C.5.1 Model equations

$$1 - \beta E_t \frac{(1 - \frac{\kappa}{2}(\pi_t - 1)^2) Y_t}{(1 - \frac{\kappa}{2}(\pi_{t+1} - 1)^2) Y_{t+1}} \frac{1}{\exp(\mu_{t+1})} \frac{R_t}{\pi_{t+1}} = 0$$

$$(1 - \eta) + \eta \left(1 - \frac{\kappa}{2}(\pi_t - 1)^2\right) Y_t + \beta \kappa E_t \frac{(1 - \frac{\kappa}{2}(\pi_t - 1)^2)}{(1 - \frac{\kappa}{2}(\pi_{t+1} - 1)^2)} (\pi_{t+1} - 1) \pi_{t+1} - \kappa (\pi_t - 1) \pi_t = 0$$

$$\left(\frac{R_{t-1}}{R_{ss}}\right)^\rho \pi_t^{(1-\rho)\psi_t} \exp(\sigma \varepsilon_t) - \frac{R_t}{R_{ss}} = 0 \quad , \quad \varepsilon_t \sim N(0, 1)$$

C.5.2 Steady state

π_t	y_t	R_t
1	$\frac{\eta-1}{\eta}$	$\frac{\exp(\mu)}{\beta} \pi_t$

C.5.3 Parameterization

β	κ	η	ρ_r	σ_r	$p_{1,1}$	$p_{2,2}$	$\mu(1)$	$\mu(2)$	$\psi(1)$	$\psi(2)$
0.9976	161	10	0.8	0.0025	0.9	0.9	0.0075	0.0025	3.1	0.9

C.6 Foerster, Rubio-Ramirez, Waggoner and Zha (2013): NK model with habits

C.6.1 Model equations

$$\frac{1}{C_t - \varphi \exp(-\bar{\mu}) C_{t-1}} - \beta \frac{\varphi}{X_{t+1} \exp(\mu_{t+1}) - \varphi C_t} - \lambda_t = 0$$

$$\frac{\beta \lambda_{t+1}}{\exp(\mu_{t+1})} \frac{R_{ss} \pi_t^{\psi_t} \exp(\sigma \varepsilon_t)}{\pi_{t+1}} - \lambda_t = 0 \quad , \quad \varepsilon_t \sim N(0, 1)$$

$$(1 - \eta) + \frac{\eta}{\lambda_t} + \beta \kappa (\pi_{t+1} - 1) \pi_{t+1} \frac{\lambda_{t+1}}{\lambda_t} \frac{X_{t+1}}{C_t} \frac{1 - \frac{\kappa}{2}(\pi_t - 1)^2}{1 - \frac{\kappa}{2}(\pi_{t+1} - 1)^2} - \kappa (\pi_t - 1) \pi_t = 0$$

$$X_t - C_t = 0$$

C.6.2 Steady state

π_t	λ_t	X_t	C_t
1	$\frac{\eta}{\eta-1}$	$\frac{\exp(\mu) - \beta \varphi}{\exp(\mu) - \varphi} \frac{\eta-1}{\eta}$	X_t

C.6.3 Parameterization

β	κ	η	ρ_r	σ	$p_{1,1}$	$p_{2,2}$	$\mu(1)$	$\mu(2)$	$\psi(1)$	$\psi(2)$	Rss	φ
0.9976	161	10	0.8	0.0025	0.9	0.9	0.0075	0.0025	3.1	0.9	$\frac{\exp(\mu)}{\beta}$	0.7

C.7 The Smets-Wouters model with switching monetary policy

We consider a medium scale DSGE model by [Smets and Wouters \(2007\)](#). We turn it into a regime switching model by introducing a Markov chain in two regimes, controlling the behavior of the parameters entering the monetary policy reaction function. In the good regime, the parameters assume the mode values estimated by [Smets and Wouters \(2007\)](#) for the constant parameter case. In the bad regime the interest rate is unresponsive to inflation and the output gap. The table below displays the parameters: r_π is the reaction to inflation; ρ , the smoothing parameter; r_y , the reaction to output gap; $r_{\Delta y}$, the reaction to the change in the output gap; $p_{good,good}$ is the probability of remaining in the good regime; $p_{bad,bad}$ is the probability of remaining in the bad regime.

	r_π	ρ	r_y	$r_{\Delta y}$	$p_{good,good}$	$p_{bad,bad}$
Good regime	1.4880	0.8762	0.0593	0.2347	0.9	NA
Bad regime	0.0000	0.0000	0.0000	0.0000	NA	0.3

Centre for Applied Macro - and Petroleum economics (CAMP)
will bring together economists working on applied macroeconomic issues, with special emphasis on petroleum economics.

BI Norwegian Business School
Centre for Applied Macro - Petroleum economics (CAMP)
N-0442 Oslo

<http://www.bi.no/camp>